

# **Echtzeit-Informationsbeschaffung in softwareintensiven Systemen**

*Entwicklung eines Frameworks  
unter Verwendung von  
Kontextinformationen und kollektiver Intelligenz*

## **Dissertation**

zur Erlangung des akademischen Grades eines  
Doktors der Wirtschaftswissenschaften

(Dr. rer. pol.)

durch die Fakultät für Wirtschaftswissenschaften der  
Universität Duisburg-Essen  
Campus Essen

vorgelegt von

**Michael Beul**  
Geldern

Essen 2017

Tag der mündlichen Prüfung: 03.04.2017

Erstgutachter: Prof. Dr. Stefan Eicker

Zweitgutachter: Prof. Dr. Volker Gruhn

# Inhaltsverzeichnis

<b>Inhaltsverzeichnis .....</b>	<b>I</b>
<b>Abbildungsverzeichnis.....</b>	<b>VI</b>
<b>Tabellenverzeichnis.....</b>	<b>IX</b>
<b>Abkürzungsverzeichnis .....</b>	<b>X</b>
<b>1 Einleitung.....</b>	<b>2</b>
1.1 Problembereich und Motivation .....	2
1.2 Zielsetzung und Forschungsfragen .....	5
1.3 Szenarios zur Verdeutlichung der Zielsetzung der Arbeit .....	7
1.3.1 Szenario 1: Aktionsbezogene und vollautomatische Informationsbereitstellung .....	7
1.3.2 Szenario 2: Explizite Informationsversorgung.....	8
1.3.3 Szenario 3: Suchoptimierung .....	9
1.3.4 Szenario 4: Erweiterte Hilfe durch Verteilung.....	11
1.4 Forschungsansatz der Arbeit .....	12
1.4.1 Evaluationsmethoden in der Wirtschaftsinformatik.....	12
1.4.2 Einordnung des verwendeten Forschungsansatzes .....	14
1.5 Aufbau der Arbeit .....	18
<b>2 Thematischer Bezugsrahmen - Grundlagen und Begriffe .....</b>	<b>21</b>
2.1 Softwareintensive Systeme .....	21
2.2 Informationsmanagement.....	23
2.2.1 Persönliches Informationsmanagement (PIM) .....	24
2.2.2 Informationsbedarf .....	25
2.2.3 Informationslogistik .....	27
2.2.4 Abgrenzung von Daten, Information und Wissen.....	29
2.3 Nutzung von Kontextinformationen.....	33
2.3.1 Kontext .....	33
2.3.2 Situation .....	35
2.3.3 Kontextsensitive Systeme .....	36
2.3.4 Ubiquitous Computing .....	39
2.3.5 Kontext- und Aktivitäts-Erkennung.....	41
2.4 Intelligenz der Masse.....	42
2.5 Überblick über verwandte Arbeiten .....	43
2.5.1 Projekt Zeitgeist .....	44

2.5.2	Growl und die Mac OS X-Mitteilungszentrale.....	46
2.5.3	Kontext-Frameworks.....	47
2.5.4	Malibu Personal Productivity Assistant.....	49
2.5.5	Yahoo Pipes.....	51
2.5.6	Google Now.....	52
2.6	Zwischenfazit.....	53
<b>3</b>	<b>Das PERCOP-Framework.....</b>	<b>57</b>
3.1	Die Zielsetzung von PERCOP.....	57
3.2	Grobentwurf von PERCOP.....	59
3.2.1	Ausgangsszenario.....	60
3.2.2	Lösungsansatz.....	63
3.2.3	Zusammenfassende Darstellung des Grobentwurfs.....	66
3.3	Architektur und Konzeption von PERCOP.....	68
3.3.1	Umgebung.....	71
3.3.2	PERCOP-Client.....	72
3.3.3	Sensoren.....	74
3.3.4	Regelsystem.....	78
3.3.5	Direkte Zuordnungen als Ergänzung zum Regelsystem.....	79
3.3.6	Information Broker.....	80
3.3.7	Crowd-Plattform.....	80
3.3.8	Informations-Repository.....	81
<b>4</b>	<b>Informationsbedarf, -beschaffung und -bereitstellung.....</b>	<b>82</b>
4.1	Grundlagen des Information Retrieval.....	82
4.1.1	Der Begriff der Relevanz.....	85
4.1.2	Retrieval-Modelle.....	86
4.1.3	Pull und Push als Strategien des Information Retrieval.....	88
4.1.4	Web Information Retrieval.....	89
4.1.5	Evaluationskriterien für IR-Systeme.....	90
4.2	Informationsbedarfsanalyse.....	91
4.3	Benutzerverhalten innerhalb des IR.....	94
4.4	Information Retrieval innerhalb von PERCOP.....	98
<b>5</b>	<b>Nutzung des Potenzials der Masse.....</b>	<b>103</b>
5.1	Kollektive Intelligenz.....	103
5.2	Crowdsourcing.....	107
5.3	Interaktive Wertschöpfung.....	110

5.4	Open Source .....	113
5.5	Potenzial der Crowd für das PERCOP-Framework.....	115
5.6	Analyse der Motivation zur Beteiligung .....	119
5.6.1	Inhaltstheorien zur Motivation .....	119
5.6.1.1	Die Bedürfnispyramide .....	119
5.6.1.2	Die ERG-Theorie.....	120
5.6.1.3	Die Zwei-Faktoren-Theorie .....	121
5.6.1.4	Die Theorie der Leistungsmotivation.....	122
5.6.2	Prozesstheorien zur Motivation.....	123
5.6.2.1	Die VIE-Theorie .....	123
5.6.2.2	Leistungsmotivation.....	124
5.6.2.3	Die Gerechtigkeitstheorie.....	124
5.6.2.4	Die Attributionstheorie.....	124
5.6.3	Intrinsische und extrinsische Motivation .....	125
5.6.4	Motivationspotenzial innerhalb von PERCOP .....	129
<b>6</b>	<b>Empfehlungssysteme und -verfahren .....</b>	<b>135</b>
6.1	Charakteristika von Empfehlungssystemen .....	135
6.2	Nutzung von Empfehlungssystemen .....	137
6.3	Empfehlungsverfahren und -algorithmen .....	138
6.3.1	Kollaboratives Empfehlen .....	139
6.3.2	Inhaltsbasiertes Empfehlen.....	142
6.3.3	Hybride Ansätze .....	145
6.3.4	Wissensbasiertes Empfehlen .....	147
6.3.5	Demographisches Empfehlen .....	148
6.3.6	Kontextbasiertes Empfehlen .....	148
6.3.7	Multidimensionale Empfehlungssysteme.....	149
6.4	Bewertungen als Kernkonzept von Empfehlungssystemen .....	151
6.5	Einflussfaktoren auf Empfehlungen .....	152
6.5.1	Datenbasis des Empfehlungssystems .....	152
6.5.2	Das „Cold Start“-Problem.....	152
6.5.3	Das „Long Tail“- Problem .....	153
6.5.4	Benutzerakzeptanz von Empfehlungen .....	154
6.5.5	Manipulation von Empfehlungen.....	156
6.6	Zusammenfassung und kritische Betrachtung von Empfehlungssystemen .....	157
6.7	Empfehlungssystem innerhalb von PERCOP .....	159
<b>7</b>	<b>Informationsquellen und -erzeugung .....</b>	<b>167</b>

7.1 Informationsquellen .....	167
7.1.1 Weblogs.....	169
7.1.2 Fragen/Antwort-Seiten (Q&A-Websites) .....	171
7.1.3 Wikis .....	172
7.1.4 Social Bookmarking .....	173
7.1.5 Microblogging .....	174
7.1.6 Weitere Informationsquellen.....	174
7.2 Kontextmodellierung und Metadatenmanagement .....	175
7.2.1 Ontologien, Taxonomien und Folksonomien .....	175
7.2.2 Semantische Annotationen auf Ressourcen-Ebene .....	180
7.2.2.1 Mikroformate.....	180
7.2.2.2 RDFa .....	181
7.2.2.3 Microdata und Schema.org .....	181
7.2.3 Metadatenmanagement in PERCOP .....	182
7.2.3.1 Semantische Annotationen.....	183
7.2.3.2 Annotationsarchitektur und -prozess .....	188
7.3 Zusammenfassung.....	191
<b>8 Prototypische Umsetzung .....</b>	<b>193</b>
8.1 Anforderungen an den Prototyp .....	193
8.2 Gesamtarchitektur des Prototyps .....	194
8.3 PERCOP-Geschäftsobjekte.....	195
8.3.1 Kontextbasierte Geschäftsobjekte .....	196
8.3.2 Allgemeine Geschäftsobjekte.....	196
8.4 PERCOP-Geschäftslogik .....	197
8.4.1 Empfehlungssystem zur Informationsermittlung .....	198
8.5 PERCOP-Web-Plattform.....	201
8.5.1 Environment Manager .....	202
8.5.2 Sensor Manager .....	203
8.5.3 Rule Manager .....	204
8.5.4 Direkte Zuordnungen (Mappings) .....	204
8.6 Sensoren .....	205
8.6.1 Generischer Sensor .....	206
8.6.2 Integrierter Sensor .....	207
8.7 Desktop-Client .....	208
8.8 PERCOP-Gesamtprozess .....	210
8.9 Zusammenfassung.....	213

<b>9 Fallbeispiele .....</b>	<b>214</b>
9.1 Motivation und Anforderungen .....	214
9.2 Fallbeispiel 1: Softwareentwicklung .....	215
9.2.1 Ausgangssituation .....	215
9.2.2 Umsetzung mit PERCOP .....	215
9.2.3 Erkenntnisgewinn .....	217
9.3 Fallbeispiel 2: Verteilte Hilfe .....	218
9.3.1 Ausgangssituation .....	218
9.3.2 Umsetzung mit PERCOP .....	218
9.3.3 Erkenntnisgewinn .....	220
9.4 Fallbeispiel 3: Bildbearbeitung .....	221
9.4.1 Ausgangssituation .....	221
9.4.2 Umsetzung mit PERCOP .....	221
9.4.3 Erkenntnisgewinn .....	222
9.5 Zusammenfassung .....	223
<b>10 Schlussbetrachtung .....</b>	<b>225</b>
10.1 Zusammenfassung .....	225
10.2 Reflexion aus der Perspektive der zugrundeliegenden Forschungsmethode ..	227
10.3 Ausblick .....	228
<b>Literaturverzeichnis .....</b>	<b>230</b>
<b>Anhang .....</b>	<b>248</b>
A.1: Analyse der Beteiligung im Rahmen von PERCOP .....	248
A.2: Crowdsourcing-Systeme im WWW .....	249
A.3: Verbindung von Ontologien innerhalb von PERCOP .....	250

# Abbildungsverzeichnis

Abbildung 1: Aufgebrachte Stunden pro Woche für Informationssuche .....	4
Abbildung 2: Gründe für die Nutzung von Blogs .....	5
Abbildung 3: Bezugspunkte für die Evaluation in der gestaltungsorientierten Wirtschaftsinformatik .....	13
Abbildung 4: Empirisch gestütztes Methodenprofil der Wirtschaftsinformatik .....	17
Abbildung 5: Aufbau der Arbeit.....	20
Abbildung 6: Zusammenhänge der Begrifflichkeiten .....	23
Abbildung 7 : Informationsbedarf und -versorgung.....	26
Abbildung 8: Einfluss des Internets auf den Informationsbedarf.....	27
Abbildung 9: Information Logistics .....	29
Abbildung 10: Wissenspyramide .....	30
Abbildung 11: Beziehung zwischen Informationen und Daten .....	31
Abbildung 12: Primäre und sekundäre Kontexttypen .....	34
Abbildung 13: Zusammenhang zwischen Kontexttypen, Entität und Situation .....	35
Abbildung 14: Klassen und Dimensionen kontextsensitiver Systeme.....	36
Abbildung 15: Layered conceptual framework for context aware systems.....	38
Abbildung 16: UbiComp Evolution Chain.....	39
Abbildung 17: UbiComp System-Modell.....	40
Abbildung 18: Architecture for processing context information.....	41
Abbildung 19: Kontext-Ebenen.....	42
Abbildung 20: Growl Benachrichtigungen (links) und Mitteilungszentrale in Mac OS X (rechts).....	46
Abbildung 21: Laufzeit-Architektur des JCAF.....	48
Abbildung 22: Technological framework for decision support .....	48
Abbildung 23: Malibu-Client „Malibu Board“ .....	50
Abbildung 24: Yahoo Pipes .....	52
Abbildung 25: Einordnung von PERCOP .....	54
Abbildung 26: Einflussmöglichkeiten der Crowd.....	65
Abbildung 27: Grobentwurf des Frameworks .....	67
Abbildung 28: PERCOP-Architektur .....	69
Abbildung 29: Layout-Skizze des PERCOP-Clients .....	74
Abbildung 30: Fuzzylogik – Linguistische Variablen .....	75
Abbildung 31: Phasen eines Fuzzylogik-Systems.....	76
Abbildung 32: PERCOP Regel-System .....	78



Abbildung 33: XML-Darstellung von Regeln .....	79
Abbildung 34: Klassisches Information Retrieval Modell .....	82
Abbildung 35: Ein allgemeines Modell zum Information Retrieval .....	83
Abbildung 36: Berrypicking.....	84
Abbildung 37: Kognitiv erweitertes Modell für Information Retrieval.....	84
Abbildung 38: Rahmenmodell für visuell direkt-manipulatives Information Retrieval .....	85
Abbildung 39: Mengen der relevanten und gefundenen Dokumente .....	86
Abbildung 40: Eine Übersicht über gängige Modelle zur Repräsentation von natürlich-sprachlichen Dokumenten.....	88
Abbildung 41: Deckung des Informationsbedarfs .....	89
Abbildung 42: Integrierte Sicht des Informationsbedarfs .....	93
Abbildung 43: Modell des Informationsverhaltens .....	95
Abbildung 44: Suchdilemma bei einem Web-IR-Prozess.....	97
Abbildung 45: Konzeptuelles Modell der Informationsrecherche .....	98
Abbildung 46: Elements of collective intelligence building blocks.....	104
Abbildung 47: Typologie von Online Creative Consumer Communities .....	106
Abbildung 48: Modell der interaktiven Wertschöpfung.....	111
Abbildung 49: Maslow's (erweiterte) Bedürfnispyramide .....	120
Abbildung 50: Fünf Quellen der Motivation.....	126
Abbildung 51: A Model for Worker's Motivation in Crowdsourcing.....	127
Abbildung 52: Empfehlungssystem innerhalb von PERCOP .....	137
Abbildung 53: Ähnlichkeitsbestimmung zwischen Elementen.....	143
Abbildung 54: Aktivierungsausbreitung .....	144
Abbildung 55: Hybrider, monolithischer Ansatz.....	145
Abbildung 56: Hybrider, paralleler Ansatz.....	146
Abbildung 57: Hybrider, sequentieller Ansatz .....	147
Abbildung 58: Multidimensionales Modell für einen „Benutzer-Element-Zeit“-Empfehlungsraum .....	150
Abbildung 59: The Long Tail.....	154
Abbildung 60: Empfehlungssystem-Taxonomie .....	158
Abbildung 61: Mehrstufiges Recommender-System.....	162
Abbildung 62: Empfehlungsprozess innerhalb von PERCOP .....	164
Abbildung 63: Klassifikationsschema von Social Software .....	168
Abbildung 64: Blogosphäre.....	170
Abbildung 65: Beispiel einer Taxonomie .....	177
Abbildung 66: Methoden zur Indexierung von Wissen .....	178
Abbildung 67: PERCOP-Ontologie.....	179

Abbildung 68: Beispiel für ein XMDP-Profil für Kontext-Informationen in PERCOP .....	183
Abbildung 69: Verwendung eines Kontext-Mikroformats.....	184
Abbildung 70: RDFa-Integration in PERCOP .....	185
Abbildung 71: PERCOP-RDF-Graph (Ausschnitt).....	187
Abbildung 72: Verbindung von Ontologien in PERCOP .....	188
Abbildung 73: Architektur einer <i>Semantic Annotation Platform</i> .....	189
Abbildung 74: Annotationsprozess von Web-Dokumenten.....	191
Abbildung 75: Kernelemente des PERCOP-Prototyps .....	195
Abbildung 76: PERCOP-Geschäftsobjekte .....	197
Abbildung 77: PERCOP-Geschäftslogik .....	198
Abbildung 78: Beziehungen und Abhängigkeiten der Geschäftsobjekte und -logik.....	201
Abbildung 79: Environment Manager des PERCOP-Prototyps .....	202
Abbildung 80: Sensor Manager des PERCOP-Prototyps.....	203
Abbildung 81: Rule Manager des PERCOP-Prototyps.....	204
Abbildung 82: Mapping-Modul des PERCOP-Prototyps .....	205
Abbildung 83: Generischer Sensor des PERCOP-Prototyps.....	207
Abbildung 84: Visual Studio-AddIn und API-Nutzung.....	208
Abbildung 85: PERCOP Desktop-Client .....	209
Abbildung 86: Regelassistent des PERCOP-Prototyps .....	210
Abbildung 87: PERCOP-Gesamtprozess.....	212
Abbildung 88: Übersicht zu Fallbeispiel 1.....	217
Abbildung 89: Übersicht zu Fallbeispiel 2.....	220
Abbildung 90: Übersicht zu Fallbeispiel 3.....	222

# Tabellenverzeichnis

Tabelle 1:	Vergleich von Design-Science-Research-Prozessen .....	15
Tabelle 2:	Eigenschaften der drei IR-Modelle.....	87
Tabelle 3:	Verfahren zur Ermittlung des Informationsbedarfs .....	92
Tabelle 4:	Abfrage-Formatierung in simulierten Suchen (Prozent) .....	96
Tabelle 5:	Crowdsourcing-Systeme im WWW .....	109
Tabelle 6:	Organisatorische Rahmenbedingungen von OSS-Projekten ....	114
Tabelle 7:	Analyse der Beteiligung im Rahmen von PERCOP .....	116
Tabelle 8:	Zusammenfassung der Erkenntnisse zur Motivation bei Next Stop Design .....	129
Tabelle 9:	Motivationsbetrachtung innerhalb von PERCOP .....	132
Tabelle 10:	Benutzer-Element-Matrix bei kollaborativen Empfehlungen .....	140
Tabelle 11:	Angepasste Datenbasis .....	142
Tabelle 12:	Recommender-Anwendungsfälle in PERCOP .....	161
Tabelle 13:	Beispiel eines PERCOP-Annotationsschemas (RDF) .....	186

# Abkürzungsverzeichnis

API .....	Application Programming Interface
BL.....	Business Logic
BLOG .....	Weblog
BO.....	Business Objects
CCI .....	Computer to Computer Interaction
CEO.....	Chief Executive Officer
CPI .....	Computer to Physical environment Interaction
CPU.....	Central Processing Unit
CS .....	Crowdsourcing
CSV .....	Comma-separated values
CVS .....	Concurrent Version System
DCMI .....	Dublin Core Metadata Initiative
DSRM.....	Design Science Research Methodology
DSS .....	Decision Support System
DTP .....	Desktop Publishing
ECA .....	Event Control Action
FAQ.....	Frequently Asked Questions
GPS .....	Global Positioning System
HCI.....	Human to Computer Interaction
HTML.....	HyperText Markup Language
ICT .....	Information and Communication Technology
ID .....	Identifier
ILOG.....	Informationslogistik
IuK .....	Information und Kommunikation
JCAF.....	Java Context Awareness Framework
IM .....	Informationsmanagement
IR .....	Information Retrieval
IS.....	Informationssystem
MD.....	Multidimensional
NN .....	Nearest Neighbour
OLAP.....	Online Analytical Processing
OS .....	Open Source
OSI .....	Open Source Initiative
OSS .....	Open Source Software
OWL.....	Web Ontology Language

---

PIM .....	Persönliches Informationsmanagement
PERCOP .....	Personal Copilot
Q&A.....	Questions and Answers
RCP .....	(Eclipse) Rich Client Platform
RDF .....	Resource Description Framework
RDFa .....	Resource Description Framework in Attributes
RDFS .....	Resource Description Framework Schema
REST.....	Representational State Transfer
RSS.....	Rich Site Summary
SDK .....	Software Development Kit
SDT .....	Self Determination Theory
SVN.....	Subversion
UbiComp ....	Ubiquitous Computing
URI.....	Uniform Resource Identifier
URL .....	Uniform Resource Locator
W3C.....	World Wide Web Consortium
WHATWG ...	Web Hypertext Application Technology Working Group
WWW.....	World Wide Web
WI .....	Wirtschaftsinformatik
XHTML.....	Extensible HyperText Markup Language
XML .....	Extensible Markup Language

# **TEIL I**

## **Einleitung und theoretischer Bezugsrahmen**

Der erste Teil leitet in die Arbeit ein, indem zunächst das Thema motiviert und der Problembereich definiert wird. Anschließend findet eine Erläuterung des theoretischen Bezugsrahmens statt.

# 1 Einleitung

Das einleitende Kapitel schildert die Motivation der vorliegenden Arbeit, definiert deren Zielsetzung und verdeutlicht diese Zielsetzung anhand ausgewählter Szenarien. Des Weiteren wird die zugrundeliegende Forschungsmethode vorgestellt und der Aufbau der Arbeit beschrieben.

## 1.1 Problembereich und Motivation

Die Bedeutung von Wissen als Faktor für wirtschaftliches und gesellschaftliches Handeln ist enorm und wächst noch weiter an. Der Austausch von Wissen, insbesondere in Form von Informationen und Daten, kennzeichnet und bestimmt die Globalisierung weit mehr als der Austausch von Waren. Ermöglicht wird diese Entwicklung durch die rasanten Fortschritte der Informations- und Kommunikationstechnik, die die Verfügbarkeit von Informationen und Daten nahezu zu jedem Zeitpunkt und an jedem Ort ermöglicht. Allerdings führt die riesige, immer weiterwachsende Menge an Informationen zu einer Informationsüberflutung (engl. *information overflow* bzw. *information overload*)<sup>1</sup>.

Die Menge der verfügbaren Informationen wird aus verschiedenen Kennzahlen ersichtlich. In diesem Zusammenhang wird vor allen Dingen auf die Bedeutung des Internets hingewiesen: Beispielsweise war die Anzahl von Webseiten im Dezember 2011 mit ca. 555 Millionen doppelt so hoch wie ein Jahr zuvor<sup>2</sup>. Im Mai 2013 wurden bereits ca. 670 Millionen aktive Webseiten gemessen - die Zuwachsrate bleibt somit auf dem extrem hohen Niveau.

Solche Zuwachsraten gelten unter anderem auch für Blogs und Tweets: Während 2006 „nur“ ca. 35 Millionen Blogs gezählt wurden, waren es 2011 schon 173 Millionen<sup>3</sup>. Die täglichen Tweets auf Twitter haben sich von 50 Millionen im Februar 2010 auf 340 Millionen im März 2012 erhöht<sup>4</sup>.

Ein wesentlicher Grund für das enorme Wachstum der verfügbaren Informationen liegt sicherlich in der Tatsache begründet, dass es sehr einfach geworden ist, Informationen als Content im Internet zu publizieren. Durch die starke Nutzung von sog. *Social Software* hat sich zudem der „Ort“ der Informationsablage verändert bzw. es sind neue Orte

---

<sup>1</sup> Siehe <http://blog.mindjet.de/?p=1245>

<sup>2</sup> Siehe <http://www.at-web.de/blog/20111212/mehr-als-555-millionen-websites-im-internet.htm>

<sup>3</sup> Siehe <http://de.statista.com/statistik/daten/studie/220178/umfrage/anzahl-der-blogs-weltweit/>

<sup>4</sup> Siehe <http://de.statista.com/statistik/daten/studie/237226/umfrage/wachstum-von-twitter-nach-anzahl-der-taeglichen-tweets/>

hinzugekommen; neben Webseiten, Blogs und Tweets sind insbesondere Wikis, Kommentare, Videos und Nachrichten auf Micro-Blogging-Plattformen zu nennen.

Die bloße Existenz der verfügbaren Informationen stiftet allerdings prinzipiell zunächst einmal keinen Nutzen; ein Nutzen ergibt sich vielmehr erst dann, wenn Informationen zum richtigen Zeitpunkt gefunden werden bzw. zur Verfügung stehen. Wegen der Durchdringung des geschäftlichen und privaten Lebens mit Anwendungssystemen bildet deren Nutzung ein sehr relevantes Beispiel: Immer mehr Personengruppen nutzen die immer weitergehenden Möglichkeiten, Informations- und Kommunikationstechnik und -funktionen zu nutzen; die Rahmenbedingungen für die Nutzung ändern sich in immer kürzeren Intervallen, weil die Systeme funktional erweitert, ihre Benutzerschnittstelle verändert und durch tiefgreifend modifizierte Versionen bisher genutzte Versionen ersetzt werden. Dies alles führt zu einem entsprechend hohen Bedarf an Informationen zur Nutzung der Systeme.

Die sich direkt anschließende Frage ist, wie Nutzer sich solche Informationen beschaffen können. Das „klassische“ Benutzerhandbuch steht vielfach gar nicht mehr zur Verfügung, ist nicht schnell genug aktualisierbar und wird auch nicht gern genutzt, insbesondere, weil die Suche nach Informationen in ihm zeitaufwendig und häufig nicht erfolgreich ist. Dafür liegen aber relevante Informationen des Anbieters und von anderen Nutzern immer umfassender „im Internet“ vor. Deshalb suchen Nutzer entsprechende Informationen dort über Suchmaschinen.

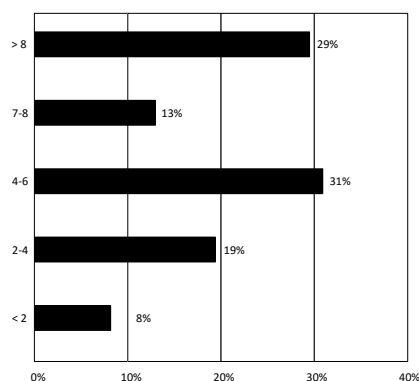
Studien belegen, dass ca. 85% aller Nutzer ihre Internet-Sitzung mit der Verwendung von Suchmaschinen beginnen (vgl. Erlhofer 2012, S. 24). Entsprechende Untersuchungen zeigen jedoch auch, dass viele Internetnutzer den Prozess der Informationssuche nicht optimal bzw. „schlecht“ durchführen, d. h. relevante Informationen nicht finden und - objektiv gesehen - ihre Informationssuche zu lange dauert.

Die Gründe für die Mängel bei der Informationssuche sind vielfältig. Erstens zeigen Studien, dass *„die meisten Nutzer [...] nicht willens [sind], bei der Formulierung ihres Suchziels allzu viel kognitive und zeitliche Energie aufzuwenden.“* (Machill 2003, S. 169). Zweitens werden die erweiterten Optionen von Suchmaschinen nur sehr selten verwendet (vgl. Spink und Jansen 2004). Drittens neigen Menschen generell dazu, nicht weiter nach alternativen (besseren) Lösungen zu suchen, wenn ihnen bereits ein Lösungsweg bekannt ist (vgl. Schmid 2004, S. 180ff., und Simon 1959, S. 253ff.). Schließlich erschwert viertens häufig die fehlende Beschreibung durch semantische Informationen das Auffinden von relevanten Informationsartefakten (vgl. Adida et al. 2012).



Im Ergebnis führt die mangelhafte Durchführung des Suchprozesses zu einem viel zu hohen Aufwand, weil die Nutzer zum einen viele Suchprozesse durchführen und zum anderen jeder Suchprozess zu viel Zeit erfordert. Somit verwundert es nicht, dass u. a. eine Studie der Delphi Group belegt, dass Mitarbeiter in Unternehmen einen (zu) hohen Anteil ihrer Arbeitszeit mit der Suche nach Informationen verbringen (siehe Abbildung 1). Im Rahmen der Studie bestätigten zudem 90% der Befragten, dass für sie die Suche nach den von ihnen benötigten Informationen schwierig und zeitaufwendig ist.

Weitere Ergebnisse der Studie sind, dass Haupteinflussfaktoren der Probleme das *Fehlen geeigneter Werkzeuge* zur Unterstützung des Suchprozesses (26%), fehlendes Bewusstsein der Nutzer für den *eigentlichen Informationsbedarf* (13%) und die *häufige Änderung der Informationen* (14%) sind. Eine andere Studie der Delphi Group hat ebenfalls diese Ergebnisse ergeben (vgl. Delphi Group 2002).



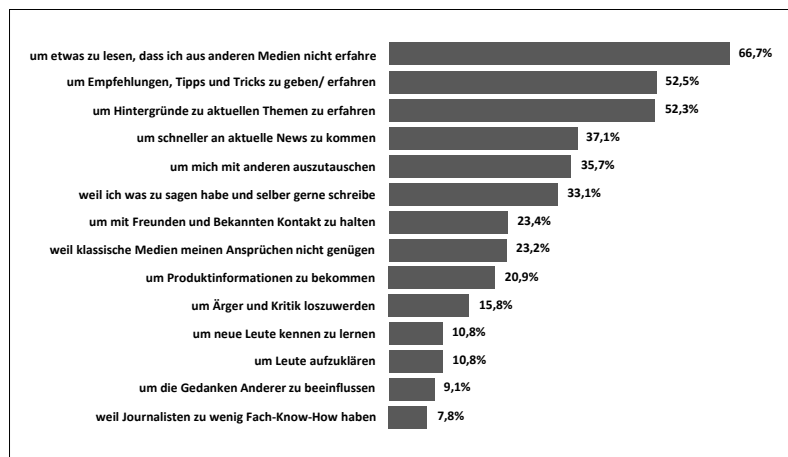
**Abbildung 1: Aufgebrachte Stunden pro Woche für Informationssuche**

Quelle: (Delphi Group 2004)

Zusammenfassend ist festzustellen, dass zum einen ein hoher Bedarf nach Informationen besteht, zum anderen sehr/zu viel Zeit für die Suche aufgewendet wird, dies vor allem deshalb, weil adäquate Werkzeuge zur Unterstützung des Suchprozesses vermisst werden.

Die mit dem *Web 2.0* entstandenen Plattformen Wikis, Foren, Blogs und Nachrichtendienste unterstützen kollaboratives und dezentrales Arbeiten. Dies bildet nicht zuletzt die Basis für einerseits OpenSource-Projekte und andererseits neue Geschäftsmodelle, bei denen sich Kunden in den Entwicklungsprozess von Produkten einbringen können. Zudem können Probleme kooperativ durch eine große Anzahl von Mitwirkenden im Rahmen der sogenannten Massenintelligenz gelöst werden; dies ist insbesondere für komplexe Probleme möglich und „interessant“ (vgl. Hippner und Wilde 2005, Bächle 2008 und O'Reilly 2005).

In diesem Kontext belegt u. a. eine Studie zur Nutzung von Blogs, welchen hohen Stellenwert *Social Software* inzwischen für die Bereitstellung und für die Suche nach Informationen besitzt (vgl. Zerfaß und Bogosyan 2007). Abbildung 2 stellt die Gründe dar, die die Studie für die Nutzung von Blogs ermittelt hat. Sie verdeutlicht, dass eine sehr große Anzahl von Benutzern Blogs verwendet, um Informationen zu konsumieren: Die vier meist genannten Gründe für die Nutzung beziehen sich auf die Suche nach nützlichen Informationen.



**Abbildung 2: Gründe für die Nutzung von Blogs**

Quelle: (Zerfaß und Bogosyan 2007)

Die Beteiligungsformen bzw. die Gründe für eine Beteiligung durch die einzelnen Individuen ergeben sich aus unterschiedlichen Aspekten der Motivation. Unterschieden werden insbesondere intrinsische und extrinsische Motivation (vgl. Deci und Ryan 1985); zu intrinsischer und extrinsischer Motivation zählen beispielsweise Spaß an der Arbeit, Zugehörigkeitsgefühl und Bestätigung, Bezahlung oder der eigene Vorteil.

## 1.2 Zielsetzung und Forschungsfragen

Die Zielsetzung der vorliegenden Arbeit besteht darin, ein Framework zu entwickeln, das Benutzern von Anwendungssystemen die Suche nach relevanten Informationen zur Nutzung eines Systems erleichtert bzw. diese Suche automatisiert durchführt; dies soll zu einer effektiveren Benutzung von Anwendungssystemen in den verschiedenen Domänen führen. Dazu soll das Framework eine kontextabhängige Echtzeit-Informationsbeschaffung „innerhalb“ der Anwendungssysteme ermöglichen, dies insbesondere auf der Basis des Potenzials der kollektiven Intelligenz.

Das Framework soll domänen- und technologieunabhängig und in diesem Sinne „generisch“ sein. Dies bedeutet, es soll eine tragfähige Basis für die Erstellung spezifischer Implementierungen bilden, die für eine oder mehrere ausgewählte Domänen mit spezifischen Technologien realisiert werden.

Bei der Entwicklung des Frameworks sollen Verfahren aus allen relevanten Wissensgebieten verwendet bzw. adaptiert werden. Neben der Wirtschaftsinformatik sind dies entsprechend der Zielsetzung der Arbeit die Bereiche *Context Awareness*, *Information Retrieval*, *Collective Intelligence* und *Semantic Web*.

Die zentrale Forschungsfrage, die sich aus der Zielsetzung ergibt, besteht somit darin, wie der Informationsbedarf eines konkreten Benutzers bei der Verwendung von Anwendungssystemen ermittelt und wie der ermittelte Informationsbedarf für Information-Retrieval-Aktivitäten genutzt werden kann.

Aus dieser zentralen Forschungsfrage ergeben sich folgende Detailfragen:

- *Informationsbedarf*: Wie lässt sich aus der Vielzahl unterschiedlicher Benutzer(-rollen) und Anwendungssysteme der konkrete Informationsbedarf ermitteln und so spezifizieren, dass eine effiziente Suche nach relevanten Informationen durchführbar ist?
- *Informationsbeschaffung*: Welche Verfahren können zur Beschaffung der Informationen eingesetzt werden?
- *Verwendung von Daten und Metadaten*: Auf welche Art und Weise müssen Daten und Metadaten erfasst, beschrieben und bereitgestellt werden, um Informationsquellen optimal für die Suche nach relevanten Informationen und ihr Auffinden aufbereiten zu können?
- *Umgebung*: Welche unterschiedlichen Typen von Umgebungen lassen sich identifizieren? Der Begriff Umgebung wird im weiteren Verlauf der Arbeit näher spezifiziert; um an dieser Stelle ein Verständnis des Lesers zu ermöglichen, sei als Beispiel für eine Umgebung eine Software-Applikation genannt.
- *Kontext*: Welche Daten werden benötigt, um domänenübergreifend einen konkreten Nutzungskontext definieren zu können; wie können diese Daten ermittelt werden?
- *Dokumenttypen*: Welche unterschiedlichen Dokumenttypen lassen sich identifizieren, und welche dieser Typen sind jeweils in welchem Nutzungskontext relevant?
- *Kollektive Intelligenz*: Wie kann das Potenzial der „Massenintelligenz“ in die unterschiedlichen Bereiche des Konzepts integriert werden?

Abzugrenzen ist die Zielsetzung der Arbeit dahingehend, dass der Ansatz darin besteht, das Verhalten bzw. das Erscheinungsbild der Anwendungssysteme selbst nicht auf der Basis der Kontextinformationen zu verändern; dies würde ja die Anwendbarkeit des Ansatzes extrem einschränken. Der Fokus der Arbeit liegt somit „allein“ auf der Beschaffung von Informationen für die effektive Nutzung gegebener Anwendungssysteme.

Die Evaluation des Frameworks und seines Grundkonzepts im Sinne der Umsetzbarkeit der Forschungsergebnisse kann zielgerichtet nur durch die Entwicklung eines entsprechenden Prototyps erfolgen. Ein solcher Proof of concept-Prototyp ist eine erste Umsetzung des Frameworks auf Basis ausgewählter Technologien für ausgewählte Anwendungsdomänen. Er demonstriert zum einen die technische Umsetzbarkeit des Frameworks, ermöglicht zum anderen die Überprüfung der Nutzbarkeit des Ansatzes über geeignete Fallbeispiele.

### **1.3 Szenarios zur Verdeutlichung der Zielsetzung der Arbeit**

Um die Zielsetzung der Arbeit und die sich aus ihr ergebenden Forschungsfragen weiter zu verdeutlichen, werden in diesem Abschnitt vier Szenarios vorgestellt. Die Szenarios werden bewusst zweck- und domänenübergreifend ausgewählt, adressieren dadurch unterschiedliche Disziplinen im Bereich des Informationsmanagements. Auf diese Disziplinen wird in Kapitel 2 näher eingegangen.

Es sei zudem darauf hingewiesen, dass in den vorgestellten Szenarios einerseits bewusster, andererseits unbewusster Informationsbedarf der Benutzer besteht. Bei unbewusstem Informationsbedarf sind sich die Benutzer keines Informationsdefizits bewusst, suchen somit von sich aus nicht aktiv nach Informationen, wie sie dies bei bewusstem Informationsbedarf - zumindest teilweise - tun.

#### **1.3.1 Szenario 1: Aktionsbezogene und vollautomatische Informationsbereitstellung**

Ausgangspunkt für dieses Szenario ist ein Benutzer, der ein oder mehrere (Software-) Anwendungssysteme verwendet: Der Benutzer führt eine bestimmte Aktion innerhalb dieser Softwareumgebung aus. In diesem Moment werden ihm „in Echtzeit“ automatisch Informationen angezeigt, die für die aktuelle Situation möglicherweise von Relevanz sind. Diese Informationen können beliebiger Art sein, beispielsweise Inhalte oder Auszüge von Webseiten, Wiki- oder Blog-Einträge, Videos oder Beiträge aus Expertensystemen.

Die Anzeige der Informationen erfolgt in einer separaten Client-Anwendung, die der Benutzer geeignet auf seinem Bildschirm platzieren kann. Er soll bei seiner aktuellen Tätigkeit nicht gestört werden, sondern vielmehr selbst entscheiden können, ob er sich die angebotenen Informationen anschauen möchte oder nicht.

In einem Beispiel könnte der Prozess folgendermaßen ablaufen:

1. Ein Benutzer verwendet eine Entwicklungsumgebung zur Softwareentwicklung und erhält eine Fehlermeldung während des Kompiliervorgangs.
2. Die Fehlermeldung wird automatisch (durch eine Komponente des Frameworks) erfasst und zusammen mit weiteren Informationen über die Umgebung und den Benutzer an die Client-Anwendung übertragen.
3. Die Client-Anwendung ermittelt daraufhin für den aktuellen Kontext relevante Informationsartefakte und stellt diese - möglichst geordnet nach entsprechenden Bewertungen - in einer Ergebnisliste dar.
4. Der Benutzer kann optional die vorgeschlagenen Informationen mit „*war hilfreich*“ oder „*war nicht hilfreich*“ bewerten.

Der skizzierte Ablauf des Szenarios impliziert komplexe Vorgänge, die in dieser Arbeit im Detail analysiert und durch das zu konzipierende Framework ermöglicht werden sollen. U. a. soll Kapitel 2 die Aspekte der Erfassung von Situationen, Kontextinformationen und Benutzeraktivitäten innerhalb unterschiedlicher Umgebungen untersuchen. Bezüglich der Ermittlung der Informationen soll in Kapitel 6 auf die verschiedenen Bereiche von Empfehlungssystemen eingegangen werden.

Für das Gesamtkonzept ist die Nutzung von kollektiver Intelligenz ein entscheidender Erfolgsfaktor. Daher soll in Kapitel 5 auf unterschiedliche Bereiche der Gruppenintelligenz, insbesondere auf Aspekte der Motivation eingegangen werden.

Das beschriebene Szenario beinhaltet alle prägenden Bestandteile des Frameworks und soll daher auch im Rahmen des Prototyps (Kapitel 8) konkret implementiert werden und bei den zur Evaluation durchzuführenden Fallbeispielen (Kapitel 9) Berücksichtigung finden.

### **1.3.2 Szenario 2: Explizite Informationsversorgung**

Die Ausgangssituation für das Szenario besteht darin, dass ein Mitarbeiter eines Unternehmens eine bestimmte Information als Hinweis angezeigt bekommen soll, sobald er eine bestimmte Datei öffnet. Folgender Ablauf ergibt sich:

1. Ein Mitarbeiter *Mit1* erstellt ein Informationsdokument *Inf1* (z. B. ein Hinweis-Dokument innerhalb des unternehmensinternen Intranets)
2. Der Mitarbeiter selbst oder ein Kollege erstellt eine Regel in der Form:  
*IF OS.Username IS ‚Mit1‘ AND OS.OpenFile IS ‚Datei1‘ THEN ShowInformation IS ‚Inf1‘*
3. Sobald der Mitarbeiter die Datei öffnet, bekommt er die passende Information angezeigt. Da die Information nur relevant ist, wenn die Datei geöffnet wird, wird sie auch nur in dieser konkreten Situation angezeigt. Dies verhindert eine unnötige Informationsverbreitung.

Das Szenario beschreibt exemplarisch die Möglichkeit der direkten Informationsversorgung und verdeutlicht dadurch insbesondere zwei Aspekte des zu entwickelnden Frameworks: Zum einen erfordert die direkte Informationsversorgung das Zusammenspiel einer Regel-Funktionalität und einer Sensorik. Eine Regel legt fest, „wann“ eine Versorgung mit bestimmten Informationen erfolgen soll; sie kann grundsätzlich beliebig komplex gestaltet werden, kann insbesondere Bezug auf die vorliegende Situation nehmen. Die Aufgabe von Sensoren ist es, durch ihre Werte bestimmte Aspekte der Situation anzuzeigen. In Kapitel 3 soll detailliert auf Regeln und Sensoren und ihre Integration in das Konzept eingegangen werden.

Zum anderen zeigt das Szenario, dass unterschiedlichen Rollen für die Beteiligung zu differenzieren sind, hier in Gestalt des Mitarbeiters, der das Dokument erstellt, ein zweiter, der es öffnet, ein dritter, der die Regel erstellt hat, und schließlich weitere, die die Sensoren realisieren. Kapitel 5 soll umfassend alle Aspekte der Beteiligung diskutieren.

### 1.3.3 Szenario 3: Suchoptimierung

Wie oben angesprochen, belegen entsprechende Studien, dass die meisten Nutzer Suchmaschinen nicht optimal nutzen; Gründe dafür sind u. a. fehlende Kenntnisse oder eine „schlechte Suchstrategie“ (vgl. Machill 2003, S. 169, Spink und Jansen 2004 und Hölscher und Strube 2000). Eine Suchstrategie ist insbesondere dann als „schlecht zu bewerten“, wenn zu wenige Suchbegriffe verwendet oder die Funktionen der Suchmaschine nicht ausreichend genutzt werden.

Ausgangssituation für das Szenario ist ein Benutzer, der die Bildbearbeitungssoftware „Adobe Photoshop“ verwendet und den Bearbeitungsfiler „Aquarell“ nutzen möchte. Der Benutzer benötigt dazu Informationen und gibt den Begriff „Aquarell“ in eine Web-Suchmaschine ein. Die Ergebnisliste der Suchmaschine liefert jedoch aufgrund des zu allgemeinen Begriffs keine sinnvollen Ergebnisse für den Benutzer.

Durch die Ermittlung und die Nutzung von Kontextinformationen können Abfragen an Suchmaschinen automatisiert formuliert und an die Maschinen gestellt werden. Die umfassende Nutzung der Kontextinformationen kann dabei die Qualität der Anfrage verbessern, da sie die Anzahl nicht relevanten Ergebnisse verringern. Eine Reduzierung der Ergebnismenge führt wiederum grundsätzlich zu einem verkürzten Suchprozess.

Die Logik der automatisierten Suche kann automatisiert und ohne Interaktion mit dem Nutzer verbessert werden, da sie dezentral (von Experten) erstellt und verwaltet werden kann.

Mögliche Aspekte bei der automatischen Erstellung von Suchabfragen sind beispielsweise die Nutzung von:

- Suchpatterns und Synonymen,
- semantischen Annotationen,
- erweiterten Suchoptionen und Operatoren sowie
- unterschiedlichen Sprachen.

Bezogen auf das beschriebene Szenario können durch die Verwendung des Frameworks automatisch Informationen des Benutzerkontexts ermittelt werden und zu einer spezifischeren Suchanfrage aggregiert werden. Beispielsweise können Informationen zum Betriebssystem, aktuell geöffneter Anwendung und der eingesetzten Version ermittelt werden. Das Framework formuliert dann z. B. automatisch die Suchanfrage „Aquarell+Filter+Photoshop+CS6+Windows“. Diese Anfrage ermöglicht es der Web-Suchmaschine, Informationen mit einer höheren Relevanz bezogen auf den konkreten Benutzerkontext zu ermitteln.

Das Szenario adressiert insbesondere die automatisierte Nutzung von Web-Suchmaschinen<sup>5</sup>; die Umsetzung des Szenarios erfordert „nur“ einen Teil des adressierten Frameworks.

Für spezielle Einsatzgebiete (wie z. B. Nicht-Standard-Anwendungen) kann auf Basis des Frameworks eine spezielle Implementierung entworfen werden, die auf die spezifischen Anforderungen angepasst werden kann und damit die Schnittstelle bildet.

---

<sup>5</sup> Auf die Aspekte des Suchverhaltens von Benutzern wird in Kapitel 4 detailliert eingegangen.

### 1.3.4 Szenario 4: Erweiterte Hilfe durch Verteilung

In diesem Szenario ist der Ausgangspunkt die Erstellung einer Benutzerdokumentation für eine Softwareanwendung, die vom Hersteller bereitgestellt und in die Anwendung (zur Verwendung durch den Benutzer) integriert wird.

Anstelle einer (häufig vorzufindenden) statischen Hilfefunktion, nutzt der Hersteller das Framework und ist dadurch in der Lage, eine dynamische und dezentrale Benutzerdokumentation bereitzustellen und somit den Umfang und die Qualität der Benutzerunterstützung jederzeit zu beeinflussen.

Folgende Aktivitäten werden dazu durchgeführt:

- Der Hersteller stellt zunächst eine Basis-Hilfe bereit (optional).
- Der Hersteller ergänzt die Softwareanwendung um Einsprung-Punkte bzw. um Meldungen über konkrete Benutzeraktivitäten wie z. B. den Aufruf einer bestimmten Funktion innerhalb der Anwendung.
- Unabhängige Entwickler entwickeln Software-Module (Sensoren<sup>6</sup>), die in der Lage sind, auf die Meldungen dieser Anwendung zu reagieren und somit Benutzeraktivitäten zu erfassen.
- Die Benutzer der Anwendung erstellen Informationsartefakte (z. B. Tutorials, Diskussionen in Foren oder Videos) und verlinken diese zu konkreten Funktionen innerhalb der Anwendung.

Hilfedokumente zu Softwareanwendungen sind häufig fest in die Anwendung integriert und werden in der Regel vom Hersteller der Anwendung erstellt. Mit Hilfe des Frameworks soll die Hilfefunktion sehr viel flexibler (dezentral und dynamisch) erfolgen; grundsätzlich ist jeder Nutzer in der Lage, zu konkreten Funktionen innerhalb einer Anwendung Informationen bereitzustellen bzw. zu verlinken (dezentral).

Diese Form der Benutzerdokumentation besitzt gegenüber der statischen (unveränderbaren) Art eine Reihe von Vorteilen: Die vom Hersteller bereitgestellte Hilfe kann im Laufe der Zeit durch eine große Anzahl von Benutzern ergänzt werden, ohne dass ein Eingriff in die Software erforderlich ist. Des Weiteren ist es möglich, den Informationsgehalt zu steigern, da auch Informationen ergänzt werden können, die für gewöhnlich nicht in

---

<sup>6</sup> Hierbei handelt es sich nicht um physikalische Sensoren, sondern um Software-Module, die in der Lage sind, Veränderungen in ihrer Umgebung wahrzunehmen. Im weiteren Verlauf der Arbeit soll detailliert auf solche Sensoren eingegangen werden.



Hilfedokumenten zu finden sind, z. B. Verweise zu Fremdhersteller-Komponenten oder Beiträge in Expertensystemen.

## 1.4 Forschungsansatz der Arbeit

### 1.4.1 Evaluationsmethoden in der Wirtschaftsinformatik

Die Evaluation der Forschungsergebnisse stellt generell einen wesentlichen Bestandteil wissenschaftlicher Forschung dar. Dies gilt nicht zuletzt für die Wirtschaftsinformatik:

*„Jenseits dedizierter Forschungsprojekte und institutioneller Rahmenbedingungen ist Evaluation gleichsam inhärenter Bestandteil wissenschaftlicher Forschung [...] Die Wirtschaftsinformatik erbt dabei gleichsam die spezifischen Evaluationsprobleme der Informatik und der Ökonomie (Aufwand, Nutzen!).“ (Frank 1998).*

Evaluation wird allgemein definiert als „systematische Tätigkeit, welche in zweck- und zielgerichteter Form die Bewertung einer Sache vornimmt“ (vgl. Riege et al. 2009, S. 72), wobei Aussagen zum Wert bzw. Nutzen eines Evaluationsgegenstandes getroffen werden (ebenda).

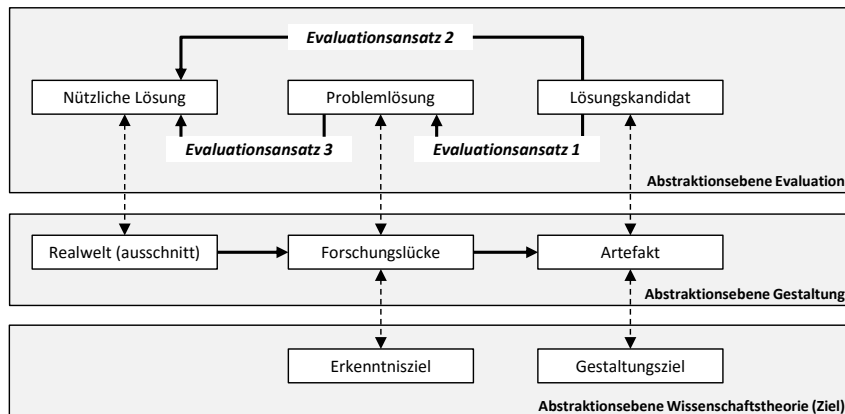
Speziell in der gestaltungs-/konstruktionsorientierten Wirtschaftsinformatik bezieht sich die Evaluation insbesondere auf den Nachweis der Nützlichkeit der erstellten Artefakte:

*„Die gestaltungsorientierte Wirtschaftsinformatik gleicht hingegen angewandten Wissenschaften wie der Medizin [...], den Ingenieurwissenschaften oder weiten Teilen der Informatik [...]. Eine zentrale Aufgabe der gestaltungsorientierten Wirtschaftsinformatik-Forschung besteht darin, die Nützlichkeit konstruierter Artefakte nachzuweisen [...].“ (Christian Fischer 2010, S. 102)*

Riege et al. unterscheiden für die gestaltungsorientierte Wirtschaftsinformatik drei Evaluationsansätze (vgl. Abbildung 3):

1. *Evaluierung des Artefakts gegen die identifizierte Forschungslücke:* Überprüfung des Artefakts in Hinblick auf korrekte Konstruktion ohne Einsatz unter Realweltbedingungen. Die Frage nach einer korrekt abgeleiteten Forschungslücke bleibt unbeantwortet.
2. *Evaluierung des Artefakts gegen die Realwelt:* Überprüfung des Artefakts gegen Realweltbedingungen. Dieser Ansatz ermöglicht die Überprüfung des Nutzens der Problemlösung, während Erkenntnisse bzgl. der Forschungslücke nur implizit erhalten werden.

3. *Evaluierung der Forschungslücke gegen die Realwelt:* Diese Form der Evaluierung spielt im Rahmen der gestaltungsorientierten WI grundsätzlich nur eine geringe Rolle.



**Abbildung 3: Bezugspunkte für die Evaluation  
in der gestaltungsorientierten Wirtschaftsinformatik**

(Riege et al. 2009, S. 75)

Riege et al. analysierten auch Evaluationsmethoden im Hinblick auf die beiden (relevanten) Ansätze „Evaluation gegen Forschungslücke“ und „Evaluation gegen Realwelt“; dabei unterscheiden sie folgende Evaluationsmethoden:

- Demonstrationsbeispiel,
- Konstruktion eines Prototyps,
- Anwendung eines Prototyps,
- Merkmalbasierter Vergleich,
- Metamodellbasierter Vergleich,
- Simulation,
- Umfrage,
- Laborexperiment,
- Feldexperiment und
- Aktionsforschung.

Ergebnis der Untersuchung ist, dass sich alle Methoden eignen, um ein Artefakt gegen eine zugrunde gelegte Forschungslücke zu evaluieren. Im Hinblick auf die Evaluation gegen die Realwelt ergab die Untersuchung, dass nur Feldexperimente, Aktionsforschung und die Anwendung von Prototypen für Evaluation geeignet sind. Simulationen und Umfragen erwiesen sich demgegenüber als nur bedingt geeignet (vgl. wieder Riege et al. 2009).

Zur detaillierten Betrachtung von Evaluationsmethoden innerhalb der gestaltungsorientierten Wirtschaftsinformatik sei an dieser Stelle auch auf die Literaturanalyse von Christian Fischer verwiesen (Christian Fischer 2010).

### 1.4.2 Einordnung des verwendeten Forschungsansatzes

Die vorliegende Arbeit ordnet sich ein in die interdisziplinäre Forschungsdisziplin *Wirtschaftsinformatik*. Das anwendungsorientierte Forschungsgebiet kombiniert insbesondere Aspekte aus der Betriebswirtschaftslehre und der Informatik zu einer Wissenschaft, die sich mit unterschiedlichen Forschungsfragen zu (betrieblichen) Anwendungs- und Informationssystemen beschäftigt (vgl. Ortner 2012 und Ardelt et al. 2011).

Nach Oesterle et al. sind der wesentliche Erkenntnisgegenstand der Wirtschaftsinformatik Informationssysteme in Wirtschaft und Gesellschaft, welche in Form von sozitechnischen Systemen Menschen, Organisationen, Informations- und Kommunikationstechnik miteinander verbinden. (vgl. Oesterle et al. 2010, S. 3)

Verfahren und Erkenntnisse verschiedener benachbarter wissenschaftlicher Disziplinen, wie z. B. die Sozialwissenschaften, die Kybernetik, die Systemtheorie und die Nachrichtentechnik, unterstützen die Forschung innerhalb der Wirtschaftsinformatik. Im Unterschied zur Mehrheit solcher benachbarter Disziplinen, bei denen die Forschungsmethoden in der Regel überwiegend theoretisch oder empirisch ausgeprägt sind, nutzt die Wirtschaftsinformatik auch andere Vorgehensweisen, da ihr Ziel sowohl Erklärung als auch Gestaltung umfasst (vgl. Heinrich 2005).

Wie bereits erwähnt, wird in Bezug auf die Forschungsmethoden in diesem Zusammenhang zwischen der *konstruktions-/gestaltungsorientierten* und der *verhaltensorientierten* Wirtschaftsinformatik unterschieden. In einem Memorandum haben bis auf wenige Ausnahmen alle Hochschullehrer der Wirtschaftsinformatik erklärt, dass dieser Forschungspluralismus bzw. „-dualismus“ für die Wirtschaftsinformatik zielführend ist (vgl. Oesterle et al. 2010 und Frank 2012).

Im Gegensatz dazu wird im angelsächsischen Bereich für die Disziplin *Information Systems Research*, die als Pendant zur Wirtschaftsinformatik gesehen wird (vgl. Oesterle et al. 2010, S. 1), in allererster Linie der verhaltensorientierte Forschungsansatz als relevant eingestuft und verfolgt. Allerdings wird auch von Vertretern der Information Systems

Research-Disziplin teilweise ein konstruktions-/gestaltungsorientierter Ansatz propagiert und beschrieben; er wird als *Design Science Research* bezeichnet<sup>7</sup>.

Tabelle 1 zeigt eine Übersicht über unterschiedliche Methoden des Design-Science-Research unterschieden nach Problemidentifikation, Lösungsentwurf und Evaluation.

	<b>Peppers et al. 2008</b>	<b>Takeda et al. 1990</b>	<b>Nunamaker et al. 1991</b>	<b>March/Smith 1995</b>	<b>Vaishnavi/Keuchler 2004/5</b>	<b>Hevner et al. 2004</b>
<b>Problem-Identifizierung</b>	Problem identification and modification  Define the objectives for a solution	Enumeration of problems	Construct a conceptual framework		Awareness of Problem	Important and relevant problems
<b>Lösungsentwurf</b>	Design and development	Suggestion  Development	Develop a system architecture  Analyze & design the system  Build the System	Build	Suggestion  Development	Implicit in „relevance“  Iterative search process  Artifact
<b>Evaluierung</b>	Demonstration  Evaluation	Evaluation to confirm the solution  Decision on a solution to be adopted	Experiment, observe & evaluate the System	Evaluate	Evaluation  Conclusion	Evaluate
<b>Kommunikation</b>	Communication					Communication

**Tabelle 1: Vergleich von Design-Science-Research-Prozessen**

Quelle: In Anlehnung an (Offermann et al. 2009, S. 4 und Peppers et al. 2007, S. 53)

<sup>7</sup> Oesterle et al. betonen, dass die Disziplin Information Systems Research im Gegensatz zur Wirtschaftsinformatik mangelnde Relevanz in der Praxis besitzt, da dort verhaltensorientierte Aspekte wie beispielsweise die „Beobachtung von Eigenschaften von Informationssystemen und des Verhaltens von Benutzern“ im Vordergrund stehen (vgl. Oesterle et al. 2010, S. 1).

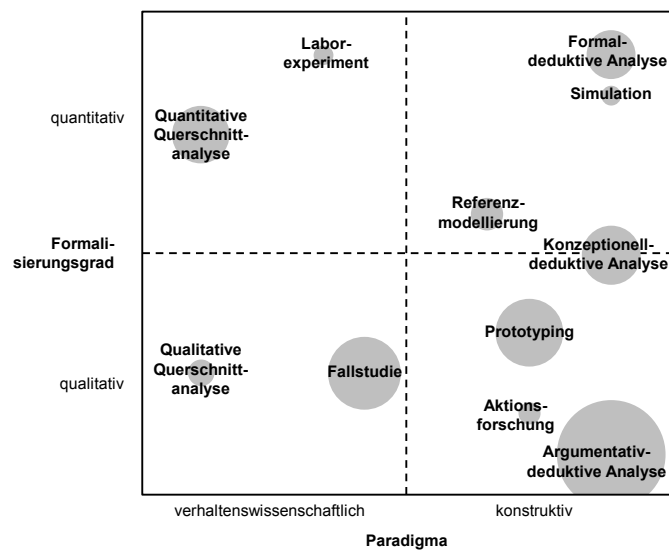
Auch Ortner betont den Unterschied: „Beiden Ausprägungen ist gemein, dass der vorherrschende Forschungsgegenstand IT-Systeme in Unternehmen und anderen menschenorientierten Gemeinschaften sind. Der Unterschied liegt im methodischen Ansatz: während die Anwendungsinformatik gestaltungsorientiert-konstruktiv und anwendungsorientiert die Entwicklung von Systemen zum Gegenstand hat [...], ist in der Information Systems eine analytische, empirisch-positivistische Forschungsweise verbreitet“ (Ortner 2012).

Tabelle 1 zeigt die einzelnen Hauptaktivitäten der Design Science Research Methodology (DSRM) von Peffers et al., die in Form von Phasen mit Iterationsschritten ablaufen. Zunächst erfolgt die Definition des konkreten Forschungsproblems, indem auf Diskussionen aus Wissenschaft und Praxis zurückgegriffen wird (Problemidentifizierung und Motivation). Anschließend folgen die Definition der Ziele sowie die sich daraus ergebenden Forschungsfragen. In den Lösungsentwurf ordnen Peffers et al. den Entwurf und die Entwicklung von Artefakten ein.

Die dieser Arbeit zugrundeliegende Forschungsmethode ist der *gestaltungsorientierten* Wirtschaftsinformatik zuzuordnen. Im Hinblick auf die Erkenntnisziele und der Ergebnistypen (Artefakte) wird im Detail die Struktur des problemzentrierten Ansatzes von Peffers et al. herangezogen:

- Die geforderten Artefakte werden innerhalb der vorliegenden Arbeit durch ein Framework und einen Prototyp repräsentiert.
- In der Evaluation-Phase sehen Peffers et al. die Beobachtung und Evaluierung der erstellten Artefakte vor. Im Rahmen der Arbeit soll dies anhand von Präsentationen des Prototyps sowie durch die Ausführung konkreter Anwendungsszenarios in Form von Fallbeispielen erfolgen.
- Zur Evaluation des in dieser Arbeit beschriebenen Lösungsansatzes werden zwei der von Riege et al. als geeignet ermittelte Methoden eingesetzt: Im ersten Schritt wird ein Prototyp konstruiert und anschließend implementiert. Dies ermöglicht in Form eines Proof of concepts Aussagen über die Realisierbarkeit der präsentierten Lösung.
- Im zweiten Schritt wird der Prototyp in mehreren Fallbeispielen eingesetzt.

Insgesamt erfolgt somit eine Evaluation sowohl im Hinblick auf die Forschungslücke als auch „gegen“ die Realwelt.



**Abbildung 4: Empirisch gestütztes Methodenprofil der Wirtschaftsinformatik**

Quelle: (Wilde und Hess 2007, S. 284)

Abbildung 4 zeigt das empirisch gestützte Methodenprofil der Wirtschaftsinformatik. Hier ordnet sich die Arbeit ein in das Feld „Formalisierungsgrad *qualitativ*, Paradigma *konstruktiv*“. Dies ergibt sich aus der Umsetzung des Prototyps und der argumentativ deduktiven Analyse. Durch die Durchführung der Fallbeispiele werden jedoch auch verhaltenswissenschaftliche Aspekte integriert.

Die Ausführungen zeigen, dass das gewählte Forschungsvorgehen den Ansprüchen des Memorandums zur gestaltungsorientierten Wirtschaftsinformatik (vgl. Oesterle et al. 2010) gerecht wird. Von den in dem Memorandum unterschiedenen *Ergebnistypen* Konstrukte, Modelle, Methoden und Instanzen werden Konstrukte und Instanzen realisiert: Die in dieser Arbeit intendierten Artefakte sind zum einen Konstrukte in Gestalt des Frameworks und zum anderen Instanzen in Gestalt des Prototyps.

Der *Erkenntnisprozess* läuft iterativ in folgenden Phasen ab:

- *Analyse:* Diese Phase entspricht der Formulierung der Problemstellung und der Forschungsziele.
- *Entwurf:* Innerhalb dieser Phase soll ein Framework und ein Prototyp entwickelt werden.
- *Evaluation:* Die Evaluation und Validierung soll einerseits durch Begutachtung der Veröffentlichungen dieses Forschungsvorhabens erfolgen, andererseits durch die Erstellung eines Prototyps und die damit umgesetzten Fallbeispiele.

- *Diffusion*: Die Kommunikation der Ergebnisse soll durch die Veröffentlichung dieser Arbeit erfolgen. Teilergebnisse wurden bereits auf wissenschaftlichen Konferenzen und Workshops publiziert und präsentiert. Der Prototyp soll in der Praxis demonstriert werden.

Im Rahmen der *Erkenntnismethoden* setzt das Forschungsvorhaben auf die Analyse von Informationssystemen, auf die Konstruktion von Demonstratoren und Prototypen sowie auf die Pilotierung und Durchführung von Fallbeispielen. Bzgl. der *Prinzipien* erfüllt die Erstellung des Frameworks die Anforderung der *Abstraktion*. *Originalität* ist ebenso gegeben, wie die nachvollziehbare *Begründung* und Validierbarkeit. Der konkrete *Nutzen* des Artefakts wird ausführlich und dies für die unterschiedlichen Anspruchsgruppen erläutert.

## 1.5 Aufbau der Arbeit

Die Arbeit wird in vier Teile strukturiert (siehe Abbildung 5). Der erste Teil leitet zunächst in die Arbeit ein (Kapitel 1). Dazu werden u. a. die Problemstellung motiviert und der zugrundeliegende Forschungsansatz erläutert. Anschließend wird in Kapitel 2 der thematische Bezugsrahmen der Arbeit erarbeitet und festgelegt. Dazu werden die wesentlichen Begriffe und Grundlagen der Bereiche *Softwaresysteme*, *Informationsmanagement*, *Kontextsensitive Systeme* und *Kollektive Intelligenz* definiert. Des Weiteren werden verwandte Arbeiten vorgestellt und diskutiert.

Im zweiten Teil der Arbeit wird entsprechend der Zielsetzung der Arbeit ein Framework zur kontextgetriebenen Informationsbeschaffung entwickelt. Der Teil besteht aus den Kapiteln 3 bis 7: Das dritte Kapitel beschreibt zunächst die Zielsetzung sowie Prämissen und Anforderungen an das Framework. Anschließend werden die Architektur sowie die Adaption und Integration unterschiedlicher Konzepte vorgestellt.

Das vierte Kapitel setzt sich dann mit dem Bedarf nach Informationen sowie der Beschaffung und der Bereitstellung von Informationen auseinander. Betrachtet werden vor allem Konzepte aus dem Bereich *Information Retrieval*.

Im fünften Kapitel wird das Potenzial der Masse untersucht. Dazu werden zunächst die Bereiche *Kollektive Intelligenz*, *Crowdsourcing*, *Interaktive Wertschöpfung* und *Open Source* analysiert. Anschließend werden auf Basis einer Motivationsanalyse die unterschiedlichen Beteiligungsmöglichkeiten innerhalb des Frameworks erarbeitet.

Gegenstand des sechsten Kapitels ist die Entwicklung eines Empfehlungssystems. Dazu werden zunächst vorhandene Empfehlungssysteme und -verfahren im Hinblick auf das

Nutzungspotenzial innerhalb des Frameworks analysiert. Anschließend wird ein Empfehlungssystem für die Integration in das Framework entworfen.

In Kapitel sieben folgt die Analyse der Informationsquellen. Zunächst werden unterschiedlichen Typen von Informationsartefakten betrachtet, die für das Framework von Relevanz sind. Anschließend wird untersucht, wie die Integration von Kontextinformationen in die Artefakte erfolgen kann. Dazu wird insbesondere auf semantische Annotationen und Ontologien eingegangen.

Der dritte Teil der Arbeit befasst sich mit der Validierung des erarbeiteten Frameworks. Kapitel 8 präsentiert einen Prototyp, der alle wesentlichen Aspekte des Frameworks umsetzt und damit die technische Realisierbarkeit belegt. Der Prototyp wird anschließend genutzt, um im Rahmen von drei ausgewählten Fallbeispielen qualitative Aussagen bezogen auf die praktische Anwendbarkeit und Adaptierbarkeit des Frameworks zu treffen (Kapitel 9).

Der vierte Teil der Arbeit bildet mit Kapitel 10 den Abschluss der Arbeit. In dem Kapitel werden zunächst die wesentlichen Ergebnisse der Arbeit zusammengefasst. Anschließend erfolgt eine Reflexion aus Sicht der Forschungsmethode.

Im Ausblick wird auf weiteren Forschungsbedarf, Erweiterungen und Einsatzgebiete des Frameworks eingegangen.



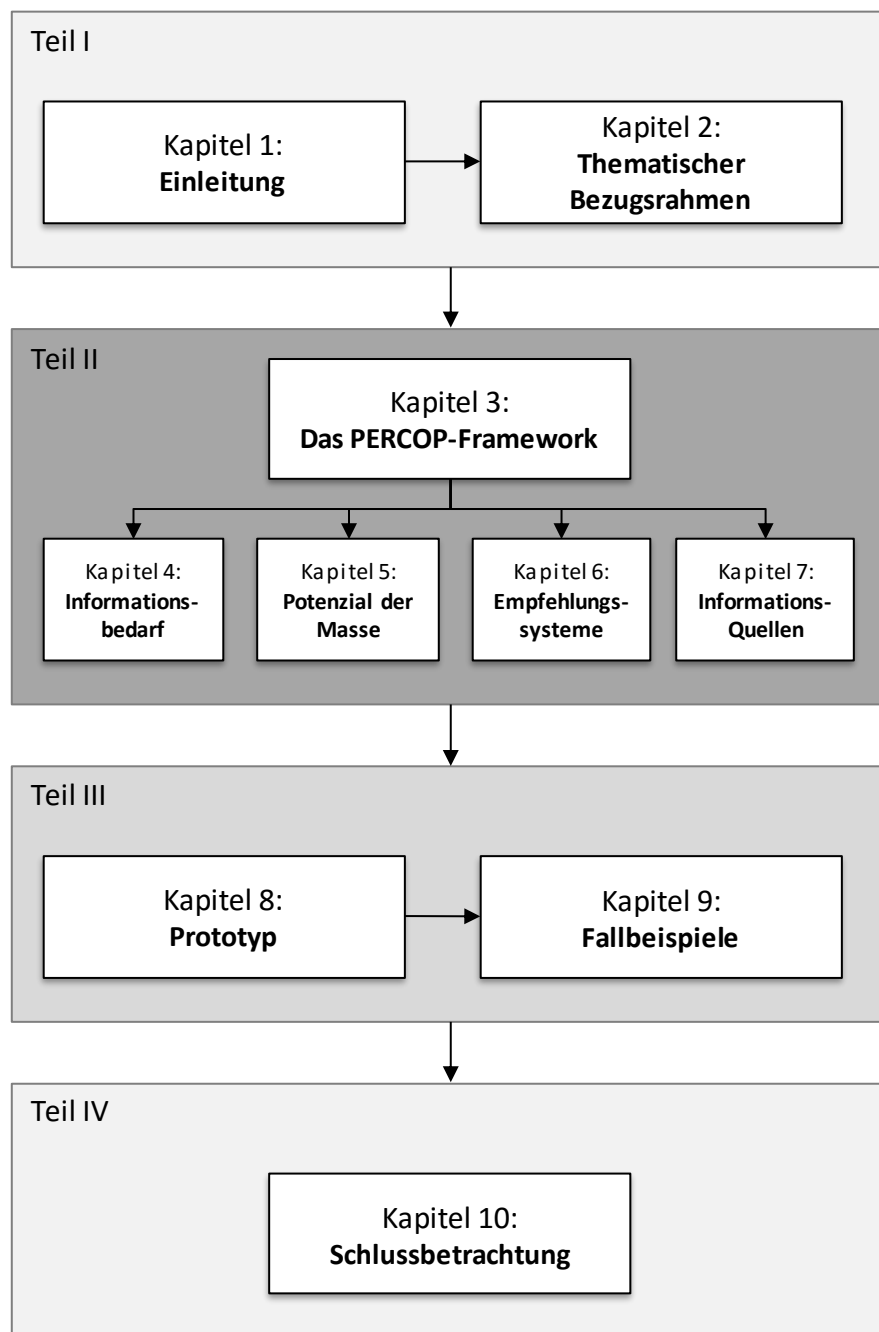


Abbildung 5: Aufbau der Arbeit

## 2 Thematischer Bezugsrahmen - Grundlagen und Begriffe

Das zweite Kapitel erarbeitet die Grundlagen der Arbeit in Gestalt der entsprechenden Themenfelder und Definitionen. Die Betrachtung erfolgt „allgemein“, d. h. unabhängig von dem intendierten Framework. Die konkrete Einbindung und Adaption in bzw. für das Framework erfolgt jeweils in den entsprechenden nachfolgenden Kapiteln.

Im Folgenden werden zunächst die Begriffe Software-, Informations- und Anwendungssysteme sowie softwareintensive Systeme diskutiert, um den für die Arbeit adäquaten Begriff für die Systeme zu identifizieren, die im Fokus des zu erarbeiteten Frameworks stehen (Abschnitt 2.1).

Anschließend wird auf die grundlegenden Aspekte des Informationsmanagements eingegangen, da die Ermittlung von Informationen das grundlegende Ziel des Frameworks bildet (Abschnitt 2.2). Abschnitt 2.3 befasst sich mit der Nutzung von Kontextinformationen, die notwendig sind, um den konkreten Informationsbedarf zu ermitteln. In Abschnitt 2.4 wird zunächst kurz auf den Bereich der Massenintelligenz eingegangen, der - wegen seiner Bedeutung für das intendierte Framework - später in Kapitel 5 sehr detailliert untersucht wird. Nach Betrachtung von verwandten Arbeiten (Abschnitt 2.5) wird ein erstes Zwischenfazit gezogen (Abschnitt 2.6).

### 2.1 Softwareintensive Systeme

Geläufige Bezeichnungen für ein System, in denen Hard- und Software durch Menschen genutzt werden, sind *Informationssystem*, *Softwaresystem* und *Anwendungssystem* bzw. *Anwendungssoftware* und *-programm*. Allerdings werden die Begriffe in Praxis und Theorie nicht einheitlich verwendet bzw. voneinander abgegrenzt. Als Beispiel sei das folgende Zitat aus der Enzyklopädie der Wirtschaftsinformatik genannt, das nicht zwischen Anwendungssystem und Softwaresystem differenziert:

*„Ein Anwendungssystem, manchmal auch als Anwendungssoftware bzw. Informationssystem bezeichnet, ist ein Softwaresystem zur Durchführung von Aufgaben in unterschiedlichen Bereichen einer Unternehmung.“* (Gabriel 2013c)

Neben der äquivalenten Verwendung der Begriffe wird in der Literatur jedoch auch häufig der Begriff des Informationssystems als Oberbegriff verwendet und damit ein Anwendungssystem als Teilsystem des Informationssystems definiert (vgl. Grob et al. 2004, S. 8). Dementsprechend kann ein Anwendungssystem verstanden werden als eine Zusammenfassung von Anwendungsprogrammen (Anwendungssoftware), „die in funktionaler und

*technischer Beziehung zueinander stehen“* (Grob et al. 2004, S. 8), mit dem Ziel, reale Aufgaben zu unterstützen.

Informationssysteme werden im engeren Sinn häufig als technische (Anwendungs-) Systeme zur Ausführung betrieblicher Aufgaben bezeichnet. Im weiteren Sinn werden auch Menschen in die Betrachtung einbezogen, was somit ein Informationssystem zu einem soziotechnischen System erweitert:

*„Ein Informationssystem (IS) ist im engeren Sinne (und so wird es i.d.R. verstanden) ein computergestütztes Anwendungssystem, d. h. ein Softwaresystem zur Ausführung betrieblicher Aufgaben. Im weiteren Sinne werden die Technik (Hard- und Software), die Menschen und die Anwendungen in einem Informationssystem zusammengefasst, das auch als Informations- und Kommunikationssystem (IuK-System) bezeichnet wird.“* (Gabriel 2013d)

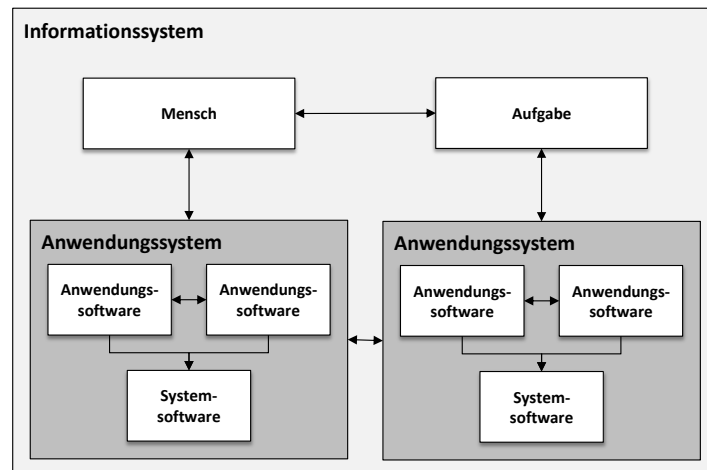
Informationssysteme lassen sich nach unterschiedlichen Kriterien klassifizieren. Insbesondere werden *strategische* und *operative* Informationssysteme unterschieden (vgl. Gabriel 2013a; Gabriel 2013b) oder auch *betriebliche* und *überbetriebliche* bzw. *branchenspezifische* und *branchenneutrale* IS (vgl. Krcmar 2010, S. 29).

Basis für die Erstellung von Informations- bzw. Anwendungssystemen ist die Software als Baustein solcher Systeme. Auch der Softwarebegriff kann im engeren und im weiteren Sinn betrachtet werden. Im engeren Sinn handelt es sich bei Software lediglich um die Programme, die auf einem Computer ausgeführt werden können. Im weiteren Sinn werden auch ergänzende Artefakte wie beispielsweise Dokumentation und andere Informationen als Bestandteil von Software verstanden (vgl. Kurbel 2014 und Kurbel 2008).

Innerhalb der Wirtschaftsinformatik wird Software häufig in *Anwendungs-* und *Systemsoftware* klassifiziert. Anwendungssoftware bezeichnet eine Software zur Unterstützung der Lösung von konkreten Anwendungsproblemen. Systemsoftware<sup>8</sup> ist Voraussetzung für die Ausführung von Anwendungssoftware. Sie repräsentiert z. B. als Betriebssystem die Verbindung zur Hardware und stellt weitere Software zur Verfügung, *„die zur Herstellung der Betriebsbereitschaft eines computerbasierten Systems erforderlich ist“* (Kurbel 2014). Abbildung 6 verdeutlicht die dieser Arbeit zugrundeliegende Beziehung der einzelnen Begriffe.

---

<sup>8</sup> Kurbel merkt an, *„dass die Verwendung und Abgrenzung der Begriffe Systemsoftware und systemnahe Software nicht einheitlich ist.“* (Kurbel 2014)



**Abbildung 6: Zusammenhänge der Begrifflichkeiten**

Als Gegenstand der Arbeit werden alle Umgebungen verstanden, die geprägt sind durch die intensive Nutzung von Software durch den Menschen, unabhängig von der eingesetzten Domäne, Technologie und der zu lösenden Aufgaben. Dies können einzelne Anwendungsprogramme sein, jedoch auch komplexe (betriebswirtschaftliche) Anwendungs- oder Informationssysteme. Im weiteren Verlauf der Arbeit wird von software-intensiven Systemen gesprochen, wobei alle zuvor erläuterten Begriffe, Systeme und Umgebungen darin eingeschlossen sind.

Der IEEE Standard 1362-1998 definiert ein solches System wie folgt:

*„A system for which software is a major technical challenge and is perhaps the major factor that affects system schedule, cost, and risk. In the most general case, a software-intensive system is comprised of hardware, software, people, and manual procedures.“ (IEEE Standards Association 2007, S. 3)*

Dieser Begriff wird bewusst gewählt, um den generischen Anspruch und das Einsatzgebiet des Frameworks zu verdeutlichen.

## 2.2 Informationsmanagement

Innerhalb der wissenschaftlichen Disziplin Wirtschaftsinformatik existieren unterschiedliche Definitionen für den Begriff Informationsmanagement (IM). Nach Stahlknecht und Hasenkamp hat das Informationsmanagement *„primär die Aufgabe, den für das Unternehmen (nach Kapital und Arbeit) ‚dritten Produktionsfaktor‘ Information zu beschaffen und in einer geeigneten Informationsstruktur bereitzustellen, und davon ausgehend die Aufgabe, die dafür erforderliche IT-Infrastruktur, d. h. die informations-technischen und personellen Ressourcen für die Informationsbereitstellung langfristig zu*

*planen und mittel- und kurzfristig zu beschaffen und einzusetzen“* (Stahlknecht und Hasenkamp 2002, S. 440).

Krcmar beschreibt innerhalb eines sog. Rahmenmodells zwei Sichtweisen des Informationsmanagements. Bei der funktionalen Sichtweise stehen Personal- und Fachfunktionen sowie die Planung, Entscheidung, Realisierung und Kontrolle im Mittelpunkt, während die institutionelle Sichtweise konkret die Personen betrachtet, welche personen- und sachbezogenen Aufgaben des IM wahrnehmen (vgl. Krcmar 2010, S. 26).

Bezugnehmend auf die Ziele des Informationsmanagements definiert Baumöl folgende Aufgaben des Informationsmanagements (Baumöl 2014a):

- Modellierung der Informationslogistik,
- Management der Schnittstelle zum Unternehmens-Controlling,
- Strategisches Informatik-Management,
- Operatives Informatik-Management und
- Qualitätsmanagement der Informatik.

Wie die beschriebenen Szenarios in Abschnitt 1.3 zeigen, betrifft das Management von Informationen im Kontext der vorliegenden Arbeit zu einem großen Teil den individuellen Nutzer bei seiner täglichen Arbeit<sup>9</sup>. Als Teilgebiet des Informationsmanagements, welches das Unternehmen als Ganzes fokussiert, beschäftigt sich das sog. *persönliche Informationsmanagement* (PIM) mit der Informationsverwaltung einzelner Anwender. Daher wird im folgenden Abschnitt näher auf das persönliche Informationsmanagement eingegangen.

## 2.2.1 Persönliches Informationsmanagement (PIM)

Als spezielle Ausprägung des IM bezeichnet das persönliche Informationsmanagement (PIM) Methoden und Vorgänge, mit denen Menschen Informationen täglich nutzen, kategorisieren und wiederverwenden (vgl. Lansdale 1988, S. 55).

*“[...] the methods and procedures by which we handle, categorize, and retrieve information on a day-to-day basis” (Lansdale 1988, S. 55)*

Teevan et al. definieren PIM etwas detaillierter als *„sowohl die Untersuchung als auch die praktische Anwendung von Tätigkeiten, die Benutzer durchführen, um ihre persönlichen*

---

<sup>9</sup> Der generische Aspekt innerhalb des erarbeiteten Frameworks ermöglicht auch die Integration in unternehmensweite Geschäftsprozesse bzw. die Nutzung innerhalb der Prozesse. Der Fokus der vorliegenden Arbeit liegt jedoch auf dem einzelnen, individuellen Benutzer.

*Informationen erfassen, organisieren, abfragen, austauschen und verwenden zu können“* (Wörndl und Schlichter 2012).

*„Personal information management (PIM) is intended to support the activities we, as individuals, perform to order our daily lives through the acquisition, organization, maintenance, retrieval, and sharing of information.“* (Teevan et al. 2006, S. 40)

Ein wesentlicher Aspekt des PIM ist die Verknüpfung von Informationen und Methoden, wobei (im Gegensatz zum allgemeinen Informationsmanagement) der Fokus auf der einzelnen Person (Benutzer) liegt. PIM ist folglich das Bindeglied zwischen den Bedürfnissen und Anforderungen von Benutzern und den benötigten Informationen, um die Bedürfnisse zu befriedigen bzw. die Aufgaben zu erledigen. Jones und Teevan unterscheiden drei Aktivitäten im Rahmen von PIM (vgl. Jones und Teevan 2007):

- Aktivitäten zur Informationserfassung,
- Aktivitäten zur Informationssuche und
- Aktivitäten zur Informationsverwaltung (Meta-Ebene).

Die Informationserfassung beinhaltet das Hinzufügen von Informationen in den persönlichen Informationsraum, wie beispielsweise das Anlegen eines Kontaktes oder eines Lesezeichens. Inhalte des sog. Informationsraums werden durch Suchaktivitäten ermittelt und durch Managementaktivitäten verwaltet. Dazu zählen beispielsweise das Anlegen von Ordnerstrukturen in einem Dateisystem oder die Definition von Kategorien innerhalb einer Email-Client-Anwendung.

Für die Ausübung der unterschiedlichen Aktivitäten wird der Benutzer durch Werkzeuge unterstützt, die sinnvollerweise miteinander integriert werden, um die Definition von Beziehungen zwischen Informationen zu ermöglichen sowie um Redundanzen zu verringern.

## 2.2.2 Informationsbedarf

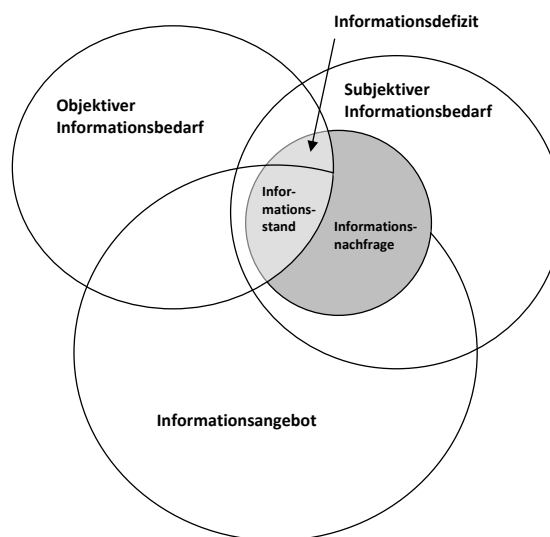
Innerhalb des Informationsmanagements ist die Ermittlung und die Deckung des Informationsbedarfs eine wesentliche Aufgabe. Eine in der Literatur häufig referenzierte Definition des Informationsbedarfs stammt von Picot et al.:

*„Der Informationsbedarf wird definiert als die Art, Menge und Qualität der Informationen, die eine Person zur Erfüllung ihrer Aufgaben in einer bestimmten Zeit benötigt.“* (Picot et al. 2003, S. 81)

Lundqvist und Sandkuhl schließen in ihrer Definition zusätzlich den Aspekt des sich ständig ändernden Bedarfs sowie den Ort der Verfügbarkeit ein:

*“Information Demand is the constantly changing need for current, accurate, and integrated information to support (business) activities, whenever and where ever it is needed.” (Lundqvist und Sandkuhl 2004)*

Picot et al. unterscheiden zwischen dem objektiven und dem subjektiven Informationsbedarf, wobei sich die Unterscheidung aus der Perspektive ergibt: Aus der Perspektive der Aufgabenstellung betrachtet der objektive Informationsbedarf die zur Erfüllung einer Aufgabe benötigten Informationen. Dem gegenüber beschreibt der subjektive Informationsbedarf aus der persönlichen Perspektive des Entscheidungsträgers, „*welche Informationen diesem zur Bewältigung einer Aufgabe als relevant erscheinen*“ (Picot et al. 2003, S. 81). Abbildung 7 zeigt den Zusammenhang zwischen den beiden Sichtweisen.



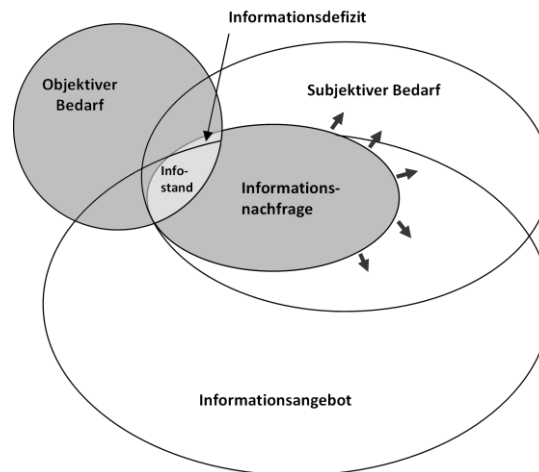
**Abbildung 7 : Informationsbedarf und -versorgung**

Quelle: In Anlehnung an (Picot et al. 2003, S. 82)

Basis für die beiden unterschiedlichen Sichtweisen der Bedarfe ist das Informationsangebot. Innerhalb des Informationsangebots bilden der subjektive und der objektive Informationsbedarf eine Schnittmenge. Die Informationsnachfrage wiederum beschreibt eine Teilmenge des gesamten Informationsbedarfs. Die konkrete Informationsversorgung entsteht dort, wo die Informationsnachfrage innerhalb des Informationsangebots liegt. Zusätzlich zur Beschaffenheit der Information bestimmt der Informationsbedarf auch Zeitpunkt, Ort und Medium der Information (vgl. Baumöl 2014b).

Durch die Entwicklung des Internets und dem damit verbundenen erhöhten Informationsangebot, hat sich auch das Verhältnis von Informationsbedarf, -nachfrage und angebot

verändert (vgl. Bartels und Pomes 2007). Insbesondere ist der subjektive Informationsbedarf im Vergleich zum objektiven Bedarf deutlich gestiegen. Abbildung 8 visualisiert die Veränderung.



**Abbildung 8: Einfluss des Internets auf den Informationsbedarf**

Quelle: (Bartels und Pomes 2007)

### 2.2.3 Informationslogistik

Da die Bereitstellung von Information in der richtigen Qualität zur richtigen Zeit am richtigen Ort ein wesentliches Ziel des intendierten Frameworks der vorliegenden Arbeit darstellt, soll im Folgenden auf den Begriff der Logistik und insbesondere der Informationslogistik näher eingegangen werden.

Mit dem Begriff Logistik wird im Allgemeinen „eine Wissenschaft, Theorie oder Methode bezeichnet, die zum Ziel hat, sicherzustellen, daß sich bestimmte Güter zum richtigen Zeitpunkt am richtigen Ort befinden. Dabei werden sowohl Güter von sehr unterschiedlicher Art berücksichtigt als auch sehr unterschiedliche Arten des Transportes.“ (Busse und Kutsche 2000, S. 1). Im Rahmen der operativen Logistik ist damit vor allem ein „effizientes Bereitstellen der geforderten Mengen benötigter Objekte in der richtigen Zusammensetzung zur rechten Zeit am richtigen Ort“ gemeint, wobei es sich bei den Objekten beispielsweise um Handelswaren, Lebensmittel, Rohstoffe, Material, Produktions- oder Betriebsmittel handeln kann (Gudehus 2011, S. 3).

Werden ausschließlich Informationen als zu verteilende Güter betrachtet, handelt es sich um das Teilgebiet Informationslogistik (ILOG). Lundqvist liefert folgende Definition des Begriffs *Informationslogistik*:



*“In short, Information Logistics is the application of information and communication technology (ICT) to a situation, organisation, or problem with the purpose of providing the right user with the right information at the right time, and to the right place. The scope of this can be an individual, a machine/facility, or any size of networked organisation.” (Lundqvist 2007, S. 3)*

Im Rahmen der Informationslogistik sind die folgenden Aspekte von besonderer Relevanz; sie werden auch als *informationslogistische Dimensionen* bezeichnet (vgl. Deiters et al. 2003, Lundqvist 2007, S. 3, und Busse und Kutsche 2000, S. 1):

- *Inhalt*: Die Informationen müssen dem konkreten Bedarf des Benutzers innerhalb seiner aktuellen Situation entsprechen. Somit muss ein ILOG-System zum einen in der Lage sein, Informationen auszuwählen, zu aggregieren und bereitzustellen. Zum anderen muss es die Fähigkeit besitzen, den Grad der Relevanz einer Information bestimmen zu können.
- *Zeit*: Um Informations-Overflow zu vermeiden, müssen Informationen genau zu der Zeit (und auch nur dann) zur Verfügung gestellt werden, wenn diese benötigt werden.
- *Ort*: Der Bedarf eines Benutzers an Informationen kann ortsabhängig sein. Ein ILOG-System muss dies ggf. berücksichtigen.
- *Präsentationsform*: Die Art und Weise, wie die Informationen übertragen und dem Benutzer präsentiert werden, muss sinnvoll von einem ILOG-System realisiert werden.

Abbildung 9 zeigt die Hauptaspekte der Informationslogistik. Lediglich die Beziehung zwischen Inhalt (Content) und Verteilung (Distribution) ist einseitig. Sowohl Inhalt als auch Verteilung besitzen eine beidseitige Abhängigkeit zum Bedarf.

Die Modellierung der Informationslogistik ist eine Teilaufgabe innerhalb des Informationsmanagements. Dabei ist die Abbildung der Entscheiderprofile, der Informationsobjekte und der Informationsflüsse ein wesentlicher Aspekt. Denn „*nur so kann eine Grundlage geschaffen werden, um die Applikationen und die Infrastruktur auf die Anforderungen der Informationslogistik auszurichten*“ (Baumöl 2014a).

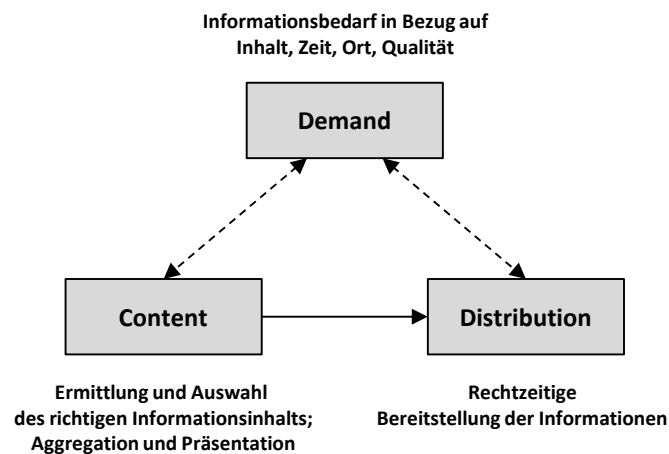


Abbildung 9: Information Logistics

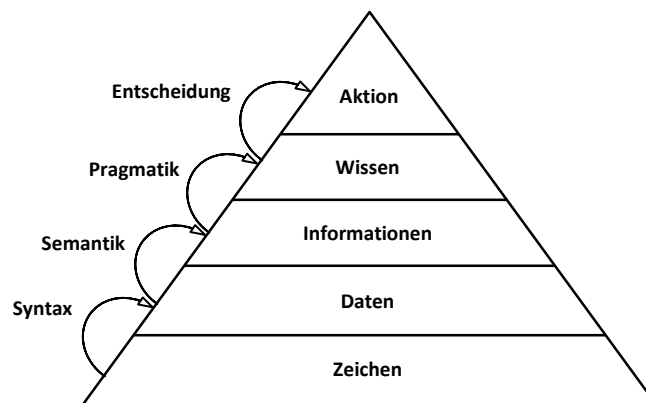
Quelle: (Sandkuhl 2009; aus dem Englischen übersetzt)

## 2.2.4 Abgrenzung von Daten, Information und Wissen

Im Rahmen des *Informations-* und *Wissensmanagements* werden insbesondere die Begriffe *Daten*, *Information* und *Wissen* verwendet, die Bedeutung dieser Begriffe ist in der Literatur jedoch nicht eindeutig definiert. Es existieren unterschiedliche Terminologien und Betrachtungsweisen, insbesondere innerhalb der Disziplinen *Informatik* und *Informationswissenschaft*. Häufig werden die Zusammenhänge der Begriffe in Form von Wissens-treppen oder Wissenspyramiden visualisiert.

Die vorliegende Arbeit orientiert sich an der Betrachtungsweise von Aamodt und Nygård, die in vielen Veröffentlichungen im Bereich der Wirtschaftsinformatik referenziert wird (vgl. Aamodt und Nygård 1995).<sup>10</sup> Abbildung 10 zeigt die Anordnung der Begriffe und die Beziehungen in Form einer Wissenspyramide nach Aamodt und Nygård.

<sup>10</sup> Explizit wird in Hinblick auf die Definitionen von *Daten*, *Information* und *Wissen* dieser Arbeit nicht die Informationsverarbeitungssicht zugrunde gelegt.



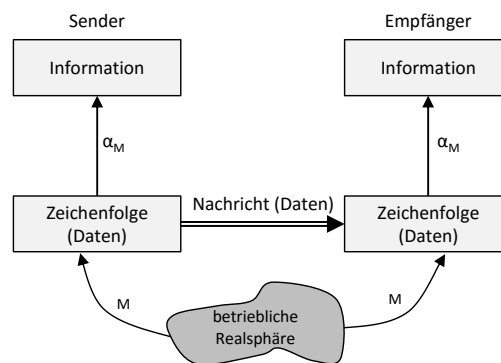
**Abbildung 10: Wissenspyramide**

Quelle: In Anlehnung an (Aamodt und Nygård 1995, S. 191)

Entsprechend der Einordnung von Aamodt und Nygård befinden sich die Zeichen auf der untersten Ebene der Hierarchie. Güldenbergs ergänzt auf dieser Ebene Signale und Reize. Aus der Menge der einzelnen Zeichen-, Signal- und Reizfolgen entstehen Daten. Nach Güldenbergs handelt es sich bei Daten um *„alle in gedruckter, gespeicherter, visueller, akustischer oder sonstiger Form verwertbaren Angaben über die verschiedensten Dinge und Sachverhalte“* (Güldenbergs 1997, S. 155). Die Daten befinden sich auf der syntaktischen Ebene und beinhalten keinerlei Semantik.

Durch die Ergänzung von Semantik lassen sich aus Daten Informationen gewinnen. In der Regel werden Daten im Rahmen von Kommunikationsprozessen zu Informationen. Nach Güldenbergs sind Informationen *„Daten, die das einzelne Individuum persönlich verwerten kann“* (Güldenbergs 1997, S. 155). Somit können Informationen als zielgerichtete, mit einer Bedeutung versehene Daten verstanden werden, wobei der Übergang von Daten zu Information durch Interpretation geschieht.

Im Gegensatz zu Daten können Informationen für unterschiedliche Akteure (Sender und Empfänger) auch eine jeweils andere Bedeutung besitzen, indem sie auf differente Weise interpretiert werden. Dieser Aspekt wird auch als empfängerorientiert bezeichnet (vgl. Güldenbergs 1997, S. 155). Die Wahrnehmung und Verwertung von Informationen geschieht folglich ausschließlich subjektiv. Die aus Daten resultierenden Informationen werden durch die verwendete Interpretationsvorschrift beeinflusst (vgl. Ferstl und Sinz 2008, S. 139). Abbildung 11 visualisiert die Zusammenhänge von Zeichen, Daten und Informationen im Rahmen der Übertragung zwischen zwei Akteuren.



**Abbildung 11: Beziehung zwischen Informationen und Daten**

Quelle: (Ferstl und Sinz 2008, S. 139)

Demnach müssen alle Akteure dieselben Interpretationsvorschriften verwenden, damit aus denselben Zeichen und Daten auch dieselben Informationen extrahiert (interpretiert) werden. Die Interpretationsvorschriften ergeben sich häufig aus der jeweiligen Situation bzw. dem jeweiligen Kontext, in dem sich die beteiligten Akteure befinden.

Auf der nächst höheren Ebene entsteht aus Informationen Wissen. In Zusammenhang mit Daten und Informationen wird Wissen in der Literatur meist als Objekt aufgefasst<sup>11</sup>. Dabei entsteht Wissen durch einen Lernprozess in Form der gedanklichen Verarbeitung von Informationen und der Verankerung in unserem Gehirn (vgl. Güldenbergs 1997). Des Weiteren stellt Wissen den Endprozess des Lernens dar und beschreibt ein subjektives Gut.

In der Betriebswirtschaftslehre wird Information als zweckorientiertes Wissen bezeichnet, wobei der Zweck in der Entscheidungsvorbereitung liegt. Innerhalb der Wirtschaftsinformatik ist Information „handlungsbestimmendes Wissen über Zustände der Wirklichkeit bzw. über Ereignisse innerhalb der Wirklichkeit“ (Heinrich und Stelzer 2011, S. 1). Eine umfassende Definition, die den Schwerpunkt auf das Individuum und die Subjektivität legt, liefern Probst et al.:

*„Wissen bezeichnet die Gesamtheit der Kenntnisse und Fähigkeiten, die Individuen zur Lösung von Problemen einsetzen. Dies umfasst sowohl theoretische Erkenntnisse als auch praktische Alltagsregeln und Handlungsanweisungen. Wissen stützt sich auf Daten und Informationen, ist im Gegensatz zu diesen jedoch immer an Personen gebunden. Es wird von Individuen konstruiert und repräsentiert deren Erwartungen über Ursache-Wirkungs-Zusammenhänge.“ (Probst et al. 1999, S. 23)*

<sup>11</sup> Neben der Betrachtungsweise als Objekt wird Wissen in der Literatur auch als Prozess oder Aktivität definiert. Vergleiche hierzu (Polanyi 1985) und (Nonaka und Takeuchi 1995).

Die Behauptung, dass die Art und Weise der Verknüpfung von Informationen zur Entstehung von (subjektiven) Wissen führt, impliziert, dass identische Informationen zu unterschiedlichem Wissen führen können (vgl. Weissenberger-Eibl und Borchers 2007). In der Literatur finden sich unterschiedliche Klassifikationen von Wissen; sie beziehen sich im Wesentlichen auf Zustand und Art des Wissens sowie auf den konkreten Wissens-träger. Unterscheiden lassen sich danach unter anderem folgende Ausprägungen von Wissen (vgl. Polanyi 1985, S. 14ff., Nonaka und Takeuchi 1995, S. 73 und Gabler Verlag 2016):

- *Implizites Wissen*: Hier handelt es sich um persönliches Wissen, welches schwer kommunizierbar ist. Beispielsweise ist Erfahrungswissen in der Regel implizites, körperliches und subjektives Wissen und bezieht sich auf gegenwärtiges Geschehen.
- *Explizites Wissen*: Wissen, welches sich in formaler und systematischer Sprache weitergeben lässt. Nonaka und Takeuchi definieren als Form des expliziten Wissens das sog. *Verstandeswissen*. Dieser Wissenstyp ist „meist explizit, metaphysisch und objektiv“ (Nonaka und Takeuchi 1995, S. 73) und bezieht sich auf vergangene Dinge und Ereignisse.
- *Individuelles Wissen*: Persönliches Wissen, bei dem die Person selbst über die Weitergabe und die Verwendung des Wissens entscheidet.
- *Kollektivwissen*: Für mehrere Personen verfügbares und nutzbares Wissen. Dazu zählen beispielsweise Gruppen oder Organisationen. Explizites Kollektivwissen wird als organisational bezeichnet.
- *Internes Wissen*: Wissen, welches innerhalb einer Organisation vorhanden ist.
- *Externes Wissen*: Wissen, das außerhalb einer Organisation existiert.
- *Metawissen*: Wissen über das Vorhandensein und die Anwendbarkeit von Wissen.

Mit Hilfe von Wissen kann eine Aktion ausgeführt werden (siehe Abbildung 10), beispielsweise das Treffen einer Entscheidung oder Tätigkeiten zur Lösung eines Problems. An einem konkreten Beispiel werden die *Zeichen* „9“ und „5“ zu dem *Datum* „95“ und zu der Information „95°C“. Diese Information führt beispielsweise unter Verwendung des *Wissens* „zu heiß“ zu der *Aktion* „Temperatur verringern“.

Aus den Ausführungen wird deutlich, dass zur Bearbeitung konkreter Aufgaben bzw. zum Treffen von Entscheidungen, Wissen notwendig ist. Wissen kann aus Informationen gewonnen werden, die mit Hilfe von Daten kommuniziert werden können, welche wiederum aus Zeichen bestehen. Um jedoch Informationen erstellen zu können, wird Wissen benötigt.

Da sowohl die Bereitstellung als auch die Nutzung von Informationen mit dem Ziel des Wissensaufbaus im Mittelpunkt der vorliegenden Arbeit stehen, werden beide Aspekte im Rahmen des intendierten Frameworks betrachtet und von diesem unterstützt. Auf die zugehörigen Prozesse wird in den Kapiteln 3, 4 und 7 näher eingegangen.

## 2.3 Nutzung von Kontextinformationen

Ziel des Frameworks ist die automatisierte Informationsversorgung eines Anwenders bezogen auf seine aktuelle Tätigkeit. Es werden daher Mechanismen und Verfahren benötigt, mit deren Hilfe auf relevante Eigenschaften der konkreten Situation bzw. des Kontexts des Anwenders geschlossen werden kann. In diesem Abschnitt wird ein kurzer Überblick über die wesentlichen Begriffe *Kontext*, *Situation* und *Kontextsensitivität* präsentiert. Im späteren Verlauf der Arbeit, werden die verschiedenen Aspekte der Kontextnutzung konkretisiert.

### 2.3.1 Kontext

Innerhalb der Sprachwissenschaften wird der Begriff *Kontext* als „Sinn- und Situationszusammenhang“ erläutert (Bibliographisches Institut GmbH). In der Literatur existiert eine Vielzahl von Definitionen für den Begriff Kontext, wobei die Definitionen häufig insbesondere Ort, Umgebung, Identität, Zeit und Situation fokussieren (vgl. z. B. Abowd und Dey 1999 und Berners-Lee et al. 2001). Eine häufig referenzierte Definition stammt von Abowd und Dey:

*“Context is any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves.” (Abowd und Dey 1999)*

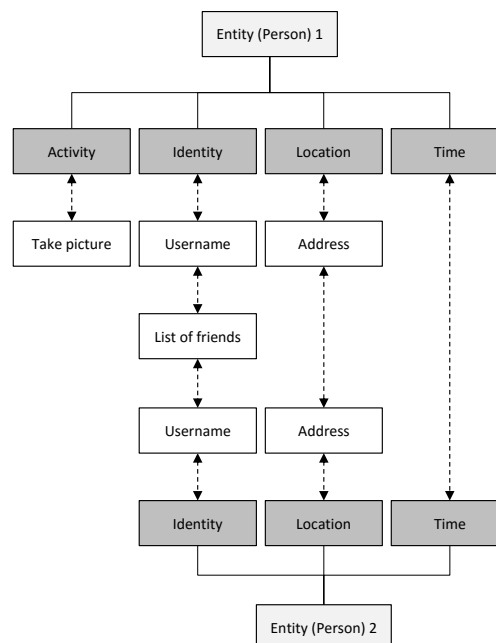
Entsprechend der Definition ist Kontext jede Information, die zur Charakterisierung der Situation einer Entität genutzt werden kann. Entitäten sind Personen, Orte oder Objekte, welche für die Interaktion zwischen einem Nutzer und einer Anwendung als relevant betrachtet werden. Der Nutzer und die Anwendung beschreiben auch Bestandteile des Kontextes, was bedeutet, dass zwei Benutzer in der gleichen Situation unterschiedliche Kontexte beschreiben.

Die Definition beinhaltet die wesentlichen für das zu erstellende Framework relevanten Bestandteile, insbesondere den Nutzer, der sich bei der Interaktion mit einer Anwendung in einer konkreten Situation befindet. Sie wird daher als Basis für die weitere Diskussion ausgewählt.

Abowd und Dey spezifizieren vier unterschiedliche Typen von Kontext (vgl. Abowd und Dey 1999):

- *Location*: Ortsinformationen, mit deren Hilfe bspw. ermittelt werden kann, welche Objekte sich in der Nähe befinden und welche Aktivitäten in derselben Umgebung ausgeführt werden.
- *Identity*: Informationen über ein Objekt bzw. über eine konkrete Person.
- *Activity*: Informationen darüber, was gerade innerhalb einer konkreten Situation „geschieht“ bzw. welche Aktivität ausgeführt wird.
- *Time*: Informationen, die den Zeitbezug herstellen.

Mit Hilfe der vier Kontexttypen kann die konkrete Situation eines Objektes spezifiziert werden; sie werden daher als *primäre Kontexttypen* bezeichnet. Alle anderen Typen von Kontext ordnen Abowd und Dey einer zweiten Ebene zu und bezeichnen sie entsprechend als *sekundäre Kontexttypen*. Sekundäre Kontextinformationen lassen sich von primären Kontextinformationen ableiten bzw. anhand von primären Kontextinformationen ermitteln; ein Beispiel zeigt Abbildung 12.



**Abbildung 12: Primäre und sekundäre Kontexttypen**

Eigene Darstellung nach (Abowd und Dey 1999)

In dem Beispiel führt eine konkrete Person (*Entity mit Identity*) eine Aktivität (*Take picture*) aus. Diese Aktivität findet an einem bestimmten Ort zu einer bestimmten Zeit statt. Aus diesen primären Kontexttypen können sekundäre Typen abgeleitet werden: Beispielsweise kann aus der Identität der Person über den zugehörigen Benutzernamen

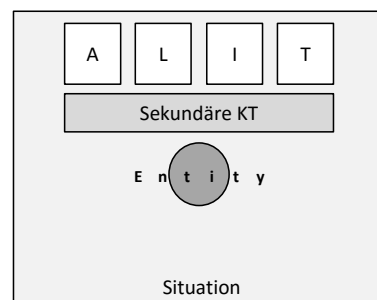
eine Liste von Freunden der Person ermittelt werden. Aus den Ortsinformationen, die beispielsweise als GPS-Koordinaten vorliegen, kann mittels *Reverse Geocoding* die zugehörige Adresse ermittelt werden. Im nächsten Schritt kann beispielsweise ermittelt werden, ob sich eine befreundete Person zur gleichen Zeit in der Nähe befindet.

### 2.3.2 Situation

Wie die bisherige Diskussion zeigt, wird häufig der Begriff „Situation“ bei der Definition und der Erläuterung von Kontext verwendet. Im Gegensatz zu den Definitionen von Pascoe und Schilit et.al (vgl. Pascoe 1998 und Schilit et al. 1994) liefern Abowd und Dey eine allgemeinere Sichtweise in Bezug zu dem Situationsbegriff:

*„Context is all about the whole situation relevant to an application and its set of users. We cannot enumerate which aspects of all situations are important, as this will change from situation to situation.“ (Abowd und Dey 1999, S. 3)*

*Situation* bezeichnet die Gesamtheit der „*Verhältnisse, Umstände, in denen sich jemand [augenblicklich] befindet*“ (Bibliographisches Institut GmbH). Übertragen auf die Kontextbetrachtung beschreibt eine Situation somit die Verhältnisse und Umstände einer Entität. Abbildung 13 zeigt den sich ergebenden Zusammenhang zwischen der Situation, der Entität und den primären und den sekundären Kontexttypen.



**Abbildung 13: Zusammenhang zwischen Kontexttypen, Entität und Situation**

A: Activity, L: Location, I: Identity, T: Time

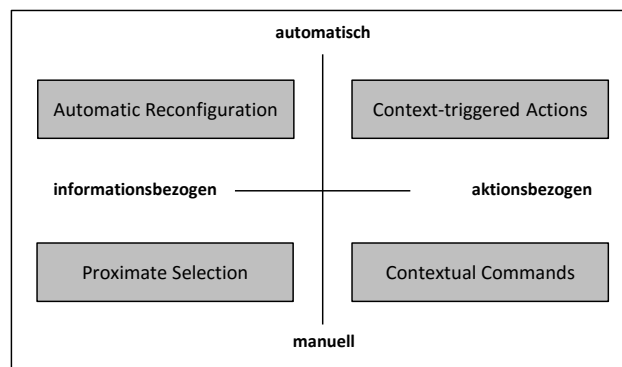
Die Herausforderung besteht in der Ermittlung der Kontextinformationen, mit denen eine Situation erkannt werden kann. Dies ist die Hauptaufgabe von sog. *kontextsensitiven Systemen*, auf die deshalb im folgenden Abschnitt näher eingegangen wird.



### 2.3.3 Kontextsensitive Systeme

Systeme, in denen die Verwendung von Kontext eine zentrale Rolle spielt, werden als *kontextsensitive Systeme*<sup>12</sup> bezeichnet. Die Bezeichnung wurde erstmalig von Schilit et al. im Jahr 1994 verwendet (vgl. Schilit et al. 1994); sie schließt sowohl die Möglichkeit, Kontext wahrzunehmen, als auch die Reaktion auf Kontextänderungen ein.

Schilit et al. teilen kontextsensitive Systeme (kontextbezogene Anwendungen) in die vier Klassen *Proximate Selection*, *Automatic Reconfiguration*, *Contextual Commands* und *Context-triggered actions* ein; die Klassen ergeben sich durch die Unterscheidung in einerseits *informations-* oder *aktionsbezogene Systeme* und andererseits in *automatische* oder *manuelle* Systeme (vgl. Abbildung 14). Informationsbezogene Anwendungen erhalten Informationen bzw. stellen diese dar, während aktionsbezogene Anwendungen Befehle ausführen. Die Ausführung der Befehle kann sowohl manuell als auch automatisch erfolgen.



**Abbildung 14: Klassen und Dimensionen kontextsensitiver Systeme**

Quelle: In Anlehnung an (Schilit et al. 1994, S. 3)

Anwendungen der Klasse *Proximate Selection* stellen dem Benutzer kontextbezogene Informationen bereit, der weitere Programmablauf geschieht manuell durch den Benutzer. Beispielsweise präsentiert eine Navigationsanwendung eine Liste möglicher Ziele, die Auswahl geschieht dann durch den Benutzer.

Die Klasse *Automatic Reconfiguration* beschreibt Anwendungen, die automatisch kontextbezogene Informationen generieren und auf z. B. verbundene Hardware übertragen. Beispielsweise könnte ein Museum automatisch raumbezogene Informationen auf die Smartphones der Besucher senden. Innerhalb eines Raumes würden alle Besucher dann dieselben Informationen erhalten.

<sup>12</sup> Als Synonyme werden auch häufig die Begriffe *Context Aware Computing* bzw. *Context Awareness* verwendet.

Werden bei einer Anwendung Befehle manuell durch den Benutzer initiiert und kontextbezogen ausgeführt, wird die Anwendung der Klasse *Contextual Commands* zugeordnet. Beispiele sind Anwendungen, die abhängig vom Ort unterschiedliche Funktionen (Befehle) bereitstellen.

Führen Anwendungen Befehle kontextbezogen und vollautomatisch (ohne manuelle Tätigkeiten durch den Benutzer) aus, werden sie als *Context-triggered* bezeichnet. Ein Beispiel ist das automatische Sperren des Computers, wenn sich der Benutzer von ihm entfernt.

Im Laufe der Zeit wurden in der Literatur andere Definitionen für kontextsensitive Systeme vorgestellt. Häufig beziehen sich die Definitionen auf spezifische Eigenschaften kontextsensitiver Systeme, wie beispielsweise Ortsinformationen, Zeit oder Temperatur (vgl. z. B. Hull et al. 1997 und Ryan et al. 1998). Da jedoch das in dieser Arbeit intendierte generische Framework technologie- und domänenunabhängig sein soll, ist die Abstützung auf eine Definition erforderlich, die kontextsensitive Systeme auf abstrakter Ebene beschreibt und somit keine Einschränkungen hinsichtlich der Umsetzung beinhaltet. Daher wird in dieser Arbeit die folgende Definition von Dey verwendet:

*„A system is context-aware if it uses context to provide relevant information and/or services to the user, where relevancy depends on the user’s task.“ (Dey 2001)*

Dey bezieht sich in seiner Definition auf die Verwendung von Kontext für die Bereitstellung von Informationen und/oder Diensten abhängig von der durchgeführten Aktivität. Seine Definition grenzt sich insbesondere von den Definitionen ab, welche in erster Linie die kontextabhängige Anpassung oder Veränderung von Software fokussieren (Schilit et al. 1994) bzw. ausschließlich orts- und personenbezogene Aspekte berücksichtigen.

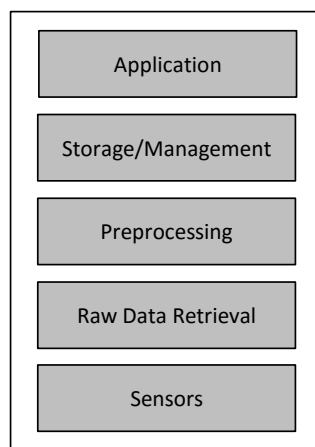
Dey definiert abhängig von ihrer Funktionalität drei Kategorien von kontextsensitiven Systemen (vgl. Dey 2001):

- *Darstellung von Informationen und Diensten:* Systeme dieser Kategorie sind in der Lage, Kontextinformationen zu ermitteln, um dem Benutzer daraufhin Informationen und Dienste zu präsentieren.
- *Ausführung von Diensten:* Systeme dieser Kategorie können automatisch Aktionen ausführen (ohne Interaktion mit dem Benutzer).
- *Kennzeichnung (tagging) von Kontext:* Systeme dieser Kategorie sind in der Lage, Kontextinformationen mit anderen Informationen durch Kennzeichnung in Verbindung zu bringen.

Des Weiteren können kontextsensitive Systeme in aktive und passive Systemtypen klassifiziert werden (vgl. Poslad 2009). In passiven Systemen wird dem Benutzer der Kontext bzw. eine Kontextänderung lediglich präsentiert; es erfolgt keine automatische Reaktion in Form einer Anpassung des Systems oder der Ausführung einer Aktion. Bei aktiven Systemen erfolgt demgegenüber eine solche Aktion; beispielsweise passt sich das System an den jeweiligen Kontext an.

Für das intendierte Framework sind insbesondere die passiven Systeme relevant<sup>13</sup>, da in erster Linie die kontextabhängige Präsentation von relevanten Informationen adressiert wird. In Bezug auf die Funktionalität sind alle drei von Dey unterschiedenen Kategorien in die Betrachtung einzubeziehen.

Die konkrete Implementierung kontextsensitiver Systeme muss abhängig von der Umgebung erfolgen, in der die Systeme jeweils Verwendung finden. Chen stellt mit der sog. *Middleware Infrastructure* einen Ansatz für die Architektur von kontextsensitiven Anwendungen in verteilten Systemen vor (vgl. Chen 2004). Bezugnehmend auf diesen Ansatz identifizieren Baldauf et al. eine allgemeine Architektur und präsentieren ein schichtenbasiertes konzeptuelles Framework für die Entwicklung kontextsensitiver Systeme (vgl. Baldauf et al. 2007, S. 3). Abbildung 15 zeigt den Aufbau des Frameworks. Auf der untersten Schicht erfassen Sensoren Werte, die zur Ermittlung des Kontexts nützlich sind. Diese Sensoren können physischer, virtueller oder logischer Art sein. Sie leiten die von ihnen erhobenen Daten (*raw data*) an die zweite Schicht weiter, die diese Daten sammelt.



**Abbildung 15: Layered conceptual framework for context aware systems**

Quelle: (Baldauf et al. 2007, S. 3)

<sup>13</sup> Es ist jedoch denkbar, das Framework in Richtung aktiver Systeme zu erweitern, indem Informationen nicht nur präsentiert werden, sondern dass aktiv in die Handlung des Benutzers eingegriffen wird.

Auf der dritten Schicht, der Preprocessing-Schicht, werden die Daten verarbeitet, um direkt nutzbare Informationen zu erhalten; beispielsweise kann mittels *Reverse Geocoding* aus GPS-Koordinaten eine konkrete Adresse ermittelt werden.

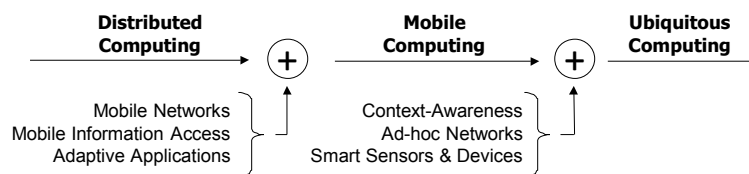
Die vierte Schicht, die Storage/Management-Schicht stellt der Applikationsschicht die gesammelten Daten und ermittelten Informationen in einer adäquat organisierten, strukturierten Form bereit. Die Applikationsschicht repräsentiert den Client, der die Kontextinformationen für eine konkrete Anwendungsfunktionalität nutzt.

### 2.3.4 Ubiquitous Computing

Insbesondere innerhalb des sog. *Ubiquitous Computing* (UbiComp) spielen kontextsensitive Systeme eine große Rolle in aktuellen Forschungsaktivitäten. 1991 von Weiser erstmalig verwendet, bezeichnet Ubiquitous Computing die Allgegenwart moderner Informationssysteme (vgl. Weiser 1991).

*„Ubiquitous Computing (UbiCom) is used to describe ICT (Information and Communication Technology) systems that enable information and tasks to be made available everywhere, and to support intuitive human usage, appearing invisible to the user.“ (Poslad 2009, S. 2)*

Nach Strang und Linnhoff-Popien (siehe Abbildung 16) stellen UbiComp-Systeme eine Spezialisierung von mobilen und verteilten Systemen dar unter Ergänzung von Kontextsensitivität, Ad-Hoc-Netzwerken sowie intelligenten Sensoren und Geräten (vgl. Strang und Linnhoff-Popien 2004).

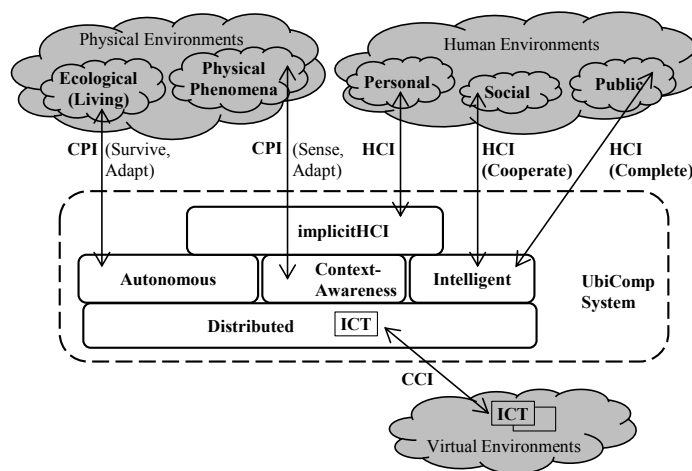


**Abbildung 16: UbiComp Evolution Chain**

Quelle: (Strang und Linnhoff-Popien 2004)

Für UbiComp-Systemen sind unterschiedliche Umgebungstypen (eng. environments) von Relevanz (siehe Abbildung 17). *Physical Environments* betreffen physikalische Dimensionen wie beispielsweise Ort, Zeit oder Temperatur. *Human Environments* beziehen sich auf die Menschen mit deren persönlichen oder sozialen Eigenschaften. *Virtual Environments* fokussieren einzelne Komponenten verteilter Systeme, die interne oder externe Dienste nutzen können (vgl. Poslad 2009, S. 14).

Über ein UbiComp-System werden die Umgebungen miteinander verbunden, wobei die Interaktionsvarianten CPI (*Computer to Physical Environment*), HCI (*Human to Computer Interaction*) und CCI (*Computer to Computer Interaction*) Verwendung finden. Die Komponenten eines UbiComp-Systems unterstützen Autonomität und Context-Awareness für CPI, intelligente Systeme sowie implizite Interaktionen für HCI und schließlich auch verteilte Systeme für CCI. Ziel eines UbiComp-Systems ist die unsichtbare (unaufdringliche) humanzentrierte Unterstützung des Benutzers durch die Nutzung von Kontextinformationen und die automatische Reaktion auf die Nutzung der Informationen (vgl. Mattern 2003, S. 4).



**Abbildung 17: UbiComp System-Modell**

Quelle: (Poslad 2009, S. 10)

Der Begriff *Pervasive Computing* wird häufig synonym zu Ubiquitous Computing verwendet. Die Grundidee der beiden Ansätze stimmt überein, da beide die Allgegenwart moderner Informationssysteme thematisieren. Der Begriff *Pervasive Computing* wurde dabei von der Industrie geprägt und mit dem Ziel verbunden, zugehörige Entwicklungen kurzfristig im Rahmen von Electronic-Commerce-Szenarien und webbasierten Geschäftsprozessen nutzbar zu machen (vgl. Mattern 2003, S. 4).

Auch der Begriff *Ambient Computing* bzw. *Ambient Intelligence* wird im Zusammenhang mit Ubiquitous Computing bzw. Pervasive Computing häufig genannt. Im Fokus von Ambient Computing steht die Verbesserung des Alltags durch die Vernetzung von Sensoren, Funk- und Computermodulen. In den Medien häufig angesprochene Beispiele sind insbesondere das *Intelligente Haus* und die Integration von Informationstechnologie in Kleidung.

### 2.3.5 Kontext- und Aktivitäts-Erkennung

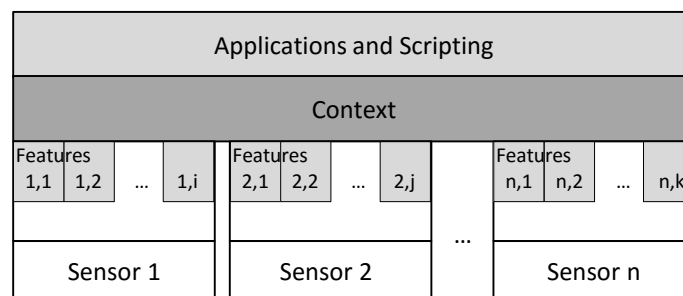
Unter Kontext-Erkennung wird der Prozess verstanden, bei dem relevante Daten aus Quellen extrahiert, aggregiert und konvertiert werden, damit sie in Anwendungen verwendet werden können. Dabei existieren Gemeinsamkeiten zur Informationsverarbeitung, insbesondere zur Signalverarbeitung, zur Mustererkennung, zur Datenanalyse sowie zum maschinellen Lernen (vgl. Mäntyjärvi 2003, S. 35).

*“Context recognition is a process for extracting, fusing and converting relevant data from sources to a representation to be utilized in the applications. [...] Thus context recognition shares common aspects with information processing in particular signal processing, pattern extraction, data analysis and machine learning.” (Mäntyjärvi 2003, S. 35)*

Bei der Kontextererkennung werden zunächst Daten aus unterschiedlichen Quellen (beispielsweise mit Hilfe von Sensoren) gesammelt. Anschließend werden die Daten aufbereitet und über verschiedene Verfahren verarbeitet, um konkrete Features zu extrahieren (*feature extraction*). Bei diesem Prozess kommen beispielsweise Methoden der Mustererkennung zum Einsatz. Die Features repräsentieren kleine Einheiten von Kontextinformationen und werden deshalb auch als *cues* oder *context atoms* bezeichnet (vgl. Mäntyjärvi 2003, S. 35, Schmidt 2002, S. 78, und Mäntyjärvi und Seppänen 2002, S. 98)<sup>14</sup>.

Mit Hilfe der identifizierten Features werden anschließend konkrete Kontextinformationen ermittelt. Dabei kommen unter anderem regelbasierte Systeme und Verfahren zum Einsatz.

Die Kontextdaten werden dann von Anwendungen genutzt. Dadurch sind Applikationen beispielsweise in der Lage, sich abhängig vom jeweiligen Kontext zu verändern. Abbildung 18 visualisiert die beschriebenen Ebenen.



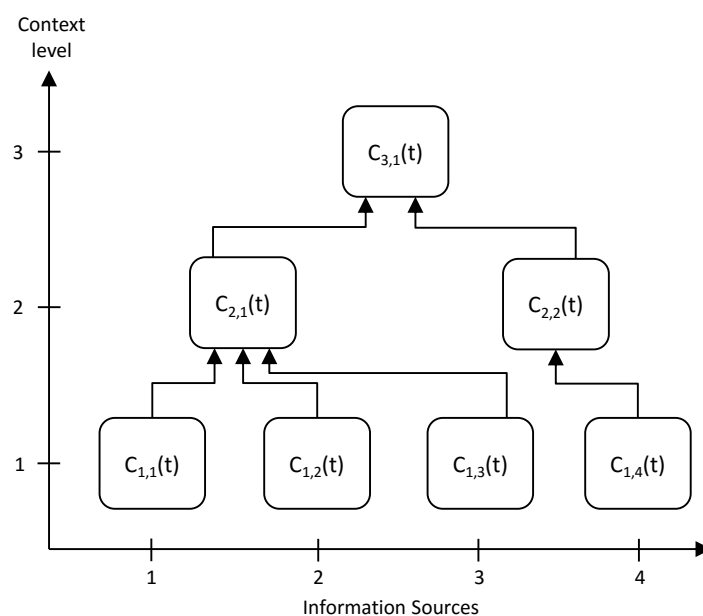
**Abbildung 18: Architecture for processing context information**

Quelle: (Mäntyjärvi 2003, S. 36) in Anlehnung an (Schmidt et al. 1999, S. 4)

<sup>14</sup> Beispielsweise können die Daten von Positions- und Beschleunigungssensoren erfasst und aus ihnen mit Hilfe von Mustererkennung die Features „Person läuft“ oder „Person fährt“ ermittelt werden.

Auch der Kontext selbst kann auf unterschiedlichen Ebenen betrachtet werden; beispielsweise können - abhängig von der Zeit und der Informationsquelle - verschiedene Kontextzustände definiert werden. Abbildung 19 verdeutlicht diesen Zusammenhang (vgl. Mäntyjärvi 2003, S. 53): Die Elemente  $C_{1,1}(t)$ ,  $C_{1,2}(t)$ ,  $C_{1,3}(t)$  und  $C_{1,4}(t)$  beschreiben Informationen auf Ebene 1. Dies können z. B. die Informationen „laufen“, „auf dem Flur“, „während der Arbeitszeit“ oder „Nutzung eines Dienstes mit einem Smartphone“ sein.

Die ersten drei Elemente werden auf Ebene zwei kombiniert zu einem Kontext  $C_{2,1}(t)$ . Dieser wiederum wird mit dem Element  $C_{1,4}(t)$  kombiniert, woraus sich der High-Level-Gesamtkontext (Ebene 3) ergibt, zum Beispiel „während der Arbeitszeit auf dem Flur laufen und einen Dienst mit einem Smartphone verwenden“.



**Abbildung 19: Kontext-Ebenen**

Quelle: (Mäntyjärvi 2003, S. 53)

## 2.4 Intelligenz der Masse

Der folgende kurze Abschnitt liefert eine Übersicht über die unterschiedlichen Konzepte zur Nutzung der Intelligenz großer Gruppen. Die konkrete Abgrenzung, der Einsatz und die Analyse der verschiedenen Ausprägungen kollektiver Intelligenz im Hinblick auf das intendierte Framework erfolgt dann im Detail in Kapitel 5.

Zum Phänomen der Massenintelligenz entstanden in den letzten Jahren insbesondere die Konzepte und Strategien *Kollektive Intelligenz*, *Crowdsourcing*, *Interaktive Wertschöpfung* und *Open Source Software*:

- *Kollektive Intelligenz*: Bezeichnet allgemein das „Zusammenbringen“ des Wissens von einzelnen Individuen, um einen positiven Nutzen für die Gesamtheit zu schaffen (vgl. Abschnitt 5.1).
- *Crowdsourcing*: Steht für die Auslagerung von ursprünglich internen (Teil-) Aufgaben an Individuen; die Individuen beteiligen sich aus unterschiedlichen Gründen an der Aufgabenerfüllung (vgl. Abschnitt 5.2).
- *Interaktive Wertschöpfung*: Bedeutet Wertschöpfung durch kooperative Zusammenarbeit, beispielsweise zwischen Kunde und Hersteller bei der Produktentwicklung (vgl. Abschnitt 5.3).
- *Open Source Software*: Bezeichnet die gemeinsame Entwicklung von Software unter Verwendung von Kollaborationswerkzeugen durch unterschiedliche Individuen (vgl. Abschnitt 5.4).

Die Aufzählung zeigt, dass die verschiedenen Konzepte und Strategien neben dem gemeinsamen Aspekt der Intelligenz großer Gruppen deutliche Unterschiede in Bezug auf Einsatz, Zielsetzung und Umsetzung aufweisen. Im weiteren Verlauf der Arbeit werden daher alle genannten Strategien und Konzepte auf ihre Nutzbarkeit im Rahmen des intendierten Frameworks hin analysiert.

## 2.5 Überblick über verwandte Arbeiten

Der folgende Abschnitt präsentiert eine Übersicht über verwandte Arbeiten. Es werden sowohl Forschungsarbeiten als auch -Projekte vorgestellt, die ähnliche Zielsetzungen wie die vorliegende Arbeit verfolgen oder vergleichbare Konzepte oder Methoden einsetzen. Darüber hinaus werden Arbeiten präsentiert, die zwar bei einer ersten Betrachtung keine Gemeinsamkeiten mit dem Thema der vorliegenden Arbeit besitzen. Jedoch sind diese Arbeiten möglicherweise für einzelne Teilaspekte von Interesse und können für einen solchen Teilaspekt einen wertvollen Beitrag zur Beantwortung der zugehörigen Forschungsfragen liefern.

Die vorgestellten Arbeiten werden in den folgenden Abschnitten auf einer vergleichsweise hohen Abstraktionsebene betrachtet. Der konkrete Bezug und die detaillierte Auseinandersetzung erfolgen dann an den jeweils relevanten Stellen der Arbeit.

Zu kontextsensitiven Systemen finden sich viele Arbeiten in den Bereichen *pervasive computing*, *mobile computing*, *ambient computing*, *ubiquitous computing*, *smart spaces*, *smart devices*, *automatically finding and recommending*.



Zudem existieren zahlreiche Veröffentlichungen zu Kontext-Ontologien, die sich mit physikalischen Umgebungen und physikalischen Sensoren beschäftigen. Beispiele für entsprechende Projekte sind *CONON* (vgl. Wang et al. 2004), *SOUPA* (vgl. Chen et al. 2004) und *CoBrA* (vgl. Chen 2004).

Des Weiteren findet sich in der Literatur eine Vielzahl von *Information Retrieval* Ansätzen, die sich mit der Suche nach relevanten Informationen beschäftigen; ein recht bekanntes Beispiel ist *POLAR* (vgl. Frommholz und Fuhr 2006).

Verwandte Arbeiten rund um den Bereich kollektiver Intelligenz oder Gruppenintelligenz finden sich beispielsweise bei *Crowdsourcing* (vgl. Howe 2008) und *Social Web Applications* (vgl. Leimeister 2010b).

Das Projekt *The KnowMore Project* befasst sich mit der kontextabhängigen proaktiven Beschaffung und Präsentation von aufgabenspezifischen Informationen innerhalb von Umgebungen, in denen ein hoher Bedarf an Wissen besteht (vgl. Abecker et al. 2000). Das primäre Ziel des Projekts in Gestalt der Echtzeit-Informationsbereitstellung unter Verwendung von Kontextinformationen gleicht dem in der vorliegenden Arbeit intendierten Konzept. Allerdings liegt der Fokus des Projekts auf der Modellierung der formalen Aufgaben und des Workflow-Kontexts. Zudem wird nicht die Einbeziehung von kollektiver Intelligenz unter besonderer Betrachtung der Beteiligungsaspekte betrachtet.

Verwandte Arbeiten in dem Themenbereich des Projekts liefern zum Beispiel Budzik und Hammond (vgl. Budzik und Hammond 2000) und Dragunov (vgl. Dragunov et al. 2005).

Im Folgenden werden einige weitere ausgewählte Projekte detaillierter vorgestellt, und es wird erläutert, welchen Bezug die Projekte jeweils zu dem in der vorliegenden Arbeit vorgestellten Framework besitzen.

### 2.5.1 Projekt Zeitgeist

Das Projekt *Zeitgeist*<sup>15</sup> hat einen Dienst entwickelt, der in der Lage ist, Benutzeraktivitäten und Ereignisse zu überwachen. Die erhobenen Informationen (z. B. geöffnete Dateien, besuchte Webseiten oder geführte Konversationen) stellt der Dienst Anwendungen zur Verfügung. Des Weiteren ist der Dienst in der Lage, mit Hilfe von Patterns Beziehungen zwischen den ermittelten Aktivitäten herzustellen.

*Zeitgeist* besteht aus folgenden Komponenten:

---

<sup>15</sup> Siehe <http://zeitgeist-project.com/about/>

- *Engine*: Dies ist das Hauptmodul. Es erhält Informationen über alle eingehenden Ereignisse und Aktivitäten, und erlaubt es Anwendungen, diese entweder abzurufen oder sich beim Hauptmodul zu registrieren, um automatisch benachrichtigt zu werden, wenn Ereignisse auftreten.
- *Extensions*: Dies sind Erweiterungen, die es erlauben, Informationen vor und nach Eingang in das Hauptmodul sowie vor dem Abruf durch Anwendungen zu manipulieren. Eine solche Manipulation kann beispielsweise eine Anreicherung der Ereignisinformationen mit zusätzlichen Daten (z. B. Geo-Daten) sein.
- *API*: Die Programmierschnittstelle erlaubt Zugriff auf die bereitgestellten Funktionen.
- *Dataproviders*: Repräsentieren Plugins für Anwendungen, welche die Benutzeraktivitäten als Ereignisse an den Zeitgeist-Dienst senden.

Zahlreiche Softwareanwendungen verwenden die Zeitgeist-Technologie, darunter unter anderem Ubuntu<sup>16</sup>, GNOME<sup>17</sup> (Activity Journal) und KDE<sup>18</sup>. Zeitgeist selbst verwendet die Technologien Vala<sup>19</sup>, Dbus<sup>20</sup> und SQLite<sup>21</sup>.

Zeitgeist ähnelt dem in der vorliegenden Arbeit intendierten Framework zum einen im adressierten Nutzer, dem Benutzer von Softwareanwendungen während der Verwendung. Zum anderen ist die Erkennung von Benutzeraktionen durch die Datenprovider sowie die Notation der erkannten Ereignisse elementarer Bestandteil sowohl von Zeitgeist als auch des in dieser Arbeit intendierten Ansatzes.

Wesentliche Unterschiede zwischen Zeitgeist und dem in dieser Arbeit intendierte Framework bestehen zum einen in Bezug auf die schwache Ausprägung der Nutzung von Massenintelligenz sowie die fehlende Kollaborationsplattform bei Zeitgeist. Zum anderen steht die Echtzeit-Informationsbeschaffung aus unterschiedlichen Informationsquellen nicht im Fokus des Zeitgeist-Projekts.

---

<sup>16</sup> Siehe <http://www.ubuntu.com/>

<sup>17</sup> Siehe <http://http://www.gnome.org/>

<sup>18</sup> Siehe <http://www.kde.org/>

<sup>19</sup> Siehe <https://live.gnome.org/Vala>

<sup>20</sup> Siehe <http://www.freedesktop.org/wiki/Software/dbus>

<sup>21</sup> Siehe <http://www.sqlite.org/>

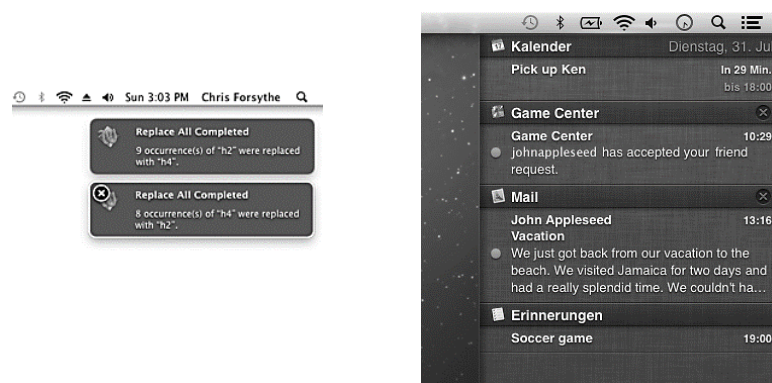
## 2.5.2 Growl und die Mac OS X-Mitteilungszentrale

Bei Growl<sup>22</sup> handelt es sich um ein globales Benachrichtigungssystem für Mac OS X und Microsoft Windows. Es stellt eine API (SDK) bereit, mit deren Hilfe Anwendungen ihre Benutzer über beliebige Ereignisse informieren können.

Das System bietet eine Reihe von Vorteilen:

- Anwendungen müssen die Benachrichtigungsfunktionalität (im Hinblick auf die Darstellung) nicht jeweils selbst implementieren.
- Die Darstellung der Benachrichtigungen ist über die Anwendungen hinweg einheitlich.
- Der Benutzer kann zentral konfigurieren, welche Benachrichtigungen er in welcher Form erhalten möchte.

Ein ähnliches Konzept wie Growl verfolgt Apple mit der seit Mac OS X 10.8 verfügbaren Mitteilungszentrale<sup>23</sup>. Auch hier können Anwendungen Nachrichten zentral ablegen. Die Funktionalität der Mitteilungszentrale ist umfassender als Growl, da sie auch die Interaktion mit dem Benutzer ermöglicht.



**Abbildung 20: Growl Benachrichtigungen (links)<sup>24</sup> und Mitteilungszentrale in Mac OS X (rechts)<sup>25</sup>**

Relevant für das intendierte Framework sind die Mitteilungssysteme in Bezug auf die Visualisierung von Nachrichten und Mitteilungen sowie zugehörige Interaktionsmöglichkeiten: Die Schnittstelle zum Endbenutzer in den Systemen kann „Input“ für die Client-Anwendung innerhalb des intendierten Frameworks geben, und eine zentrale API zur

<sup>22</sup> Siehe <http://growl.info>

<sup>23</sup> Siehe [http://support.apple.com/kb/ht5362?viewlocale=de\\_de&locale=de\\_de](http://support.apple.com/kb/ht5362?viewlocale=de_de&locale=de_de)

<sup>24</sup> Siehe [http://growl.info/images/screenshots\\_images/MultipleDisplays.png](http://growl.info/images/screenshots_images/MultipleDisplays.png)

<sup>25</sup> Siehe [http://support.apple.com/kb/HT2477?viewlocale=de\\_DE&locale=de\\_DE](http://support.apple.com/kb/HT2477?viewlocale=de_DE&locale=de_DE)

Übertragung von beliebigen Ereignisinformationen kann eine Alternative im Rahmen der konkreten Umsetzung des Konzepts darstellen.

### 2.5.3 Kontext-Frameworks

Für die Erstellung kontextsensitiver Applikationen und Systeme existiert eine Reihe von Frameworks. Beispielsweise liefern das *Java Context Awareness Framework* (JCAF) und das *ILOG-Framework* Input für das intendierte Framework im Hinblick auf die Nutzung von Kontextinformationen und der sich daraus ergebenden Situationserkennung. Daher wird auf die beiden Frameworks im Folgenden näher eingegangen.

JCAF ist ein Framework zur Erstellung von kontextsensitiven Applikationen auf Basis von Java (vgl. Bardram 2005). Die wesentlichen Elemente von JCAF sind *Context Services*, *Entities* und *Context*: Ein Context Service empfängt, verwaltet, speichert und verteilt Kontextinformationen über die Entities.

Entities repräsentieren Objekte, über die Kontextinformationen verwaltet werden sollen, beispielsweise Personen oder Orte.

Context Clients repräsentieren die kontextsensitiven Applikationen und können Kontextinformationen senden und auf Veränderungen von Kontextinformationen reagieren. Um aktuelle Kontextinformationen zu erhalten, können sich Context Clients bei Context Events registrieren.

Über den Context Service können Context Clients Interesse an Ereignissen konkreter Entitäten anmelden. Dadurch erhalten die Clients Benachrichtigungen bei Eintreten eines Ereignisses. Abbildung 21 visualisiert die Laufzeit-Architektur des JCAF.

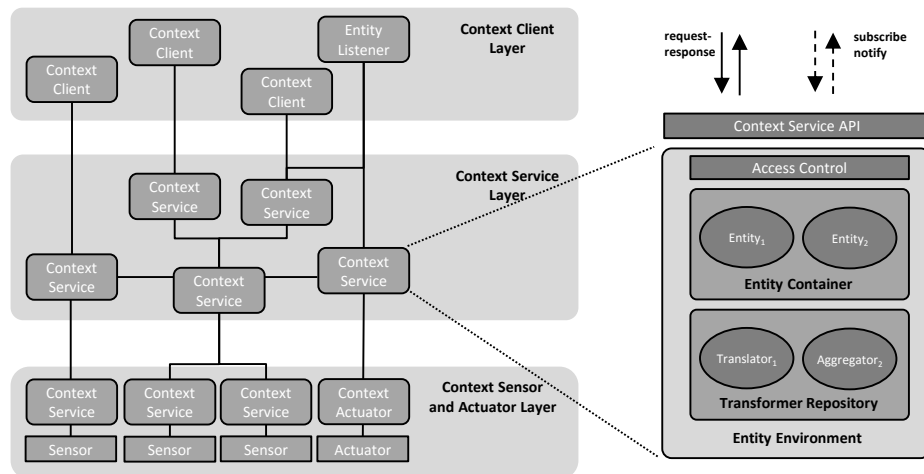


Abbildung 21: Laufzeit-Architektur des JCAF

Quelle: (Bardram 2005, S. 101)

Mit dem *Technological framework for decision support* präsentieren Levashova et al. ein Framework für die Entscheidungsunterstützung durch kontextabhängige Informationsbereitstellung (vgl. Levashova et al. 2006). Der Schwerpunkt bei diesem Framework liegt auf der Problemorientierung.

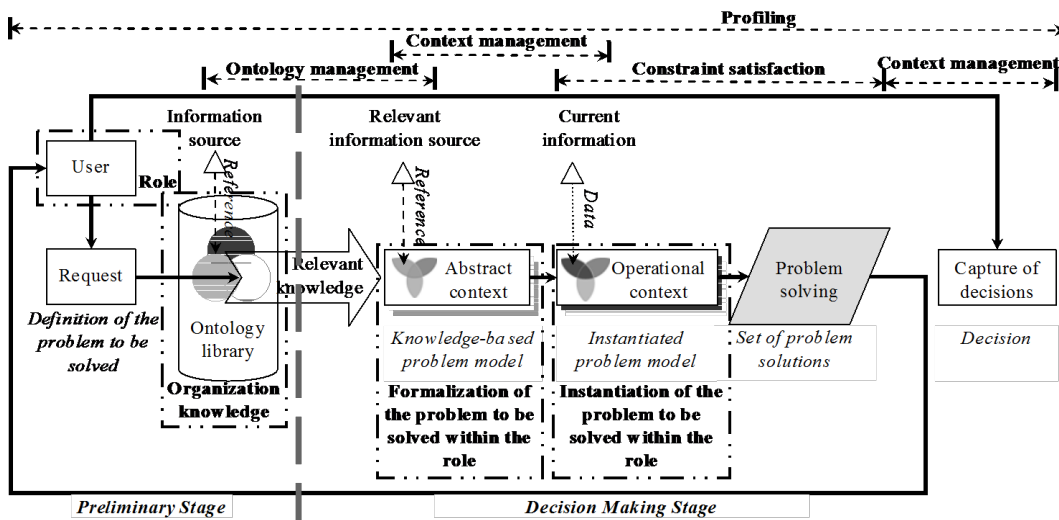


Abbildung 22: Technological framework for decision support

Quelle: (Levashova et al. 2006, S. 6)

Der in Abbildung 22 dargestellte Ansatz ist in zwei Phasen aufgeteilt, der *Preliminary Stage* und der *Decision Making Stage*. Innerhalb der ersten Phase werden initiale, vorbereitende Maßnahmen des DSS (*Decision Support System*) durchgeführt. Darunter fallen beispielsweise das Erstellen von semantischen Modellen für Informationsquellen, Domänenwissen und Benutzer sowie die Verbindung von Domänenwissen mit der Informationsquelle.

In der Phase *Decision Making Stage* erfolgt dann die Integration von Informationen und Wissen, welche für die Problemlösung und -modellierung relevant sind. Den Ausgangspunkt für die Phase stellt die formelle Beschreibung des Problems dar. Mit Hilfe von Verwaltungstechnologien für Ontologien wird auf dieser Basis relevantes Wissen gesucht<sup>26</sup>.

Frameworks wie die dargestellten Beispiele können einen Beitrag für das intendierte Framework liefern; dies betrifft zum einen – insbesondere bezogen auf das *JCAF* - technische Aspekte und zum anderen – insbesondere für das *Technological framework for decision support* - konzeptuelle Teillösungen.

Unterschiede zwischen den verfügbaren Frameworks und dem intendierten Framework bestehen vor allen Dingen darin, dass die Frameworks nicht die Massenintelligenz einbeziehen. Außerdem soll in der vorliegenden Arbeit nicht ausschließlich ein konkretes Framework entwickelt werden, sondern ein Framework für die Erstellung von konkreten Frameworks (Meta-Framework) und deren Implementierungen. Die Grenze zwischen Framework und Implementierung ist an dieser Stelle als „fließend“ zu betrachten, da eine konkrete Implementierung auch wieder ein Framework repräsentieren kann.

## 2.5.4 Malibu Personal Productivity Assistant

Ziel des Forschungsprojekts *Malibu Personal Productivity Assistant* am IBM Cambridge Research Center<sup>27</sup> ist die Unterstützung von *Knowledge Workern* bei ihrer täglichen Arbeit; ihnen sollen relevante Informationen für ihre aktuell durchzuführenden Aufgaben zur Verfügung gestellt werden. Das entwickelte Verfahren wird als *activity centric resource recommendation* bezeichnet; es nutzt automatisierte Techniken des semantischen Webs, um jeweils relevante Informationen bereitstellen zu können

Ein im Rahmen des Projekts entwickelter Prototyp setzt auf dem Activity Management System *Lotus Activities* von IBM Lotus auf, erweitert dieses um eine Client-Anwendung, die die Schnittstelle zu den Endbenutzern bildet. Abbildung 23 zeigt die Oberfläche dieses sog. Malibu-Clients.

---

<sup>26</sup> Detaillierte Informationen über das Framework finden sich bei (Levashova et al. 2006).

<sup>27</sup> Siehe <http://www.watson.ibm.com/cambridge/Projects/project4.shtml>

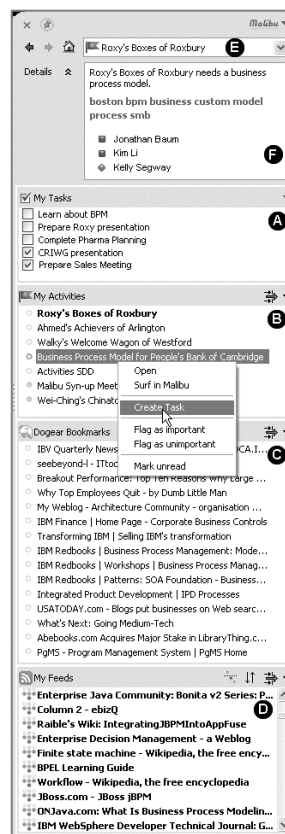


Abbildung 23: Malibu-Client „Malibu Board“

Quelle: (IBM-Research<sup>28</sup>)

Der Prototyp stellt Informationen aus diversen Datenquellen in Form von Sichten (Views) dar, z. B. Aufgaben (A), Aktivitäten (B), Bookmarks (C) und Feeds (D). Die Informationen innerhalb der verschiedenen Sichten sind abhängig von dem aktuell ausgewählten Malibu-Element (E), dessen Details in der darunterliegenden Sicht angezeigt werden (F).

Für das in der vorliegenden Arbeit intendierte Framework sind zum einen die unterschiedlichen verwendeten Informationsquellen und die Mapping-Algorithmen interessant. Zum anderen können Verfahren der Kontextermittlung und Konzepte der grafischen Repräsentation aufgegriffen werden.

Unterschiede zum Ansatz der vorliegenden Arbeit bestehen insbesondere in der fehlenden Integration der Crowd sowie in Bezug auf die in der Arbeit adressierte spezifische Zielgruppe in Gestalt der Software-Anwender.

<sup>28</sup> Siehe <http://www.watson.ibm.com/cambridge/Projects/project4.shtml>

## 2.5.5 Yahoo Pipes

Yahoo Pipes<sup>29</sup> ist ein Werkzeug, mit dessen Hilfe sich Daten aus unterschiedlichen Quellen zu einer Ausgabe kombinieren lassen. Dazu können Datenquellen wie beispielsweise RSS-Feeds, CSV- oder XML-Dateien auf einer Arbeitsfläche angeordnet und verknüpft werden. Des Weiteren besteht die Möglichkeit der Verwendung von Filter-, Sortier- und Übersetzungsfunktionen sowie von komplexeren Modulen, die beispielsweise reguläre Ausdrücke auswerten oder HTML-Dateien parsen.

*"Pipes is a powerful composition tool to aggregate, manipulate, and mashup content from around the web. [...] Pipes is a free online service that lets you remix popular feed types and create data mashups using a visual editor. You can use Pipes to run your own web projects, or publish and share your own web services without ever having to write a line of code"*<sup>30</sup>

Abbildung 24 zeigt die Verknüpfung zweier Feeds mit anschließender Sortierung nach dem Publikationsdatum. Das Ergebnis selbst kann wieder als Input für eine andere Pipe genutzt werden.

Im Vergleich mit dem in der vorliegenden Arbeit intendierten Framework bestehen Gemeinsamkeiten in Bezug auf die Aggregation von Informationsquellen sowie auf die Verfügbarkeit und Bereitstellung der Ergebnisse. Dazu liefert Yahoo Pipes zum einen Ideen und Lösungsansätze zur Verknüpfung unterschiedlicher Datenquellen. Zum anderen können die erstellten Pipes auch als Datenquelle innerhalb einer auf Basis des intendierten Frameworks erstellten Implementierung eingebunden werden. Zudem könnte ein Editor zur Unterstützung der Erstellung von Konfigurationen im Rahmen von Empfehlungen (siehe Abschnitte 6.7 und 8.4.1) sinnvoll sein, der eine ähnliche Funktionalität wie der Yahoo Pipes Editor bietet.

Aufgrund des ausschließlichen auf die Aggregation von Informationsquellen beschränkten Funktionsumfangs von Yahoo Pipes, besitzen Yahoo Pipes und das Framework keine weiteren Gemeinsamkeiten. Jedoch können möglicherweise Funktionen von Yahoo Pipes sinnvoll in das intendierte Framework eingebunden werden.

---

<sup>29</sup> Siehe <http://pipes.yahoo.com/pipes/>

<sup>30</sup> Siehe <http://pipes.yahoo.com/pipes/> und <http://pipes.yahoo.com/pipes/docs>



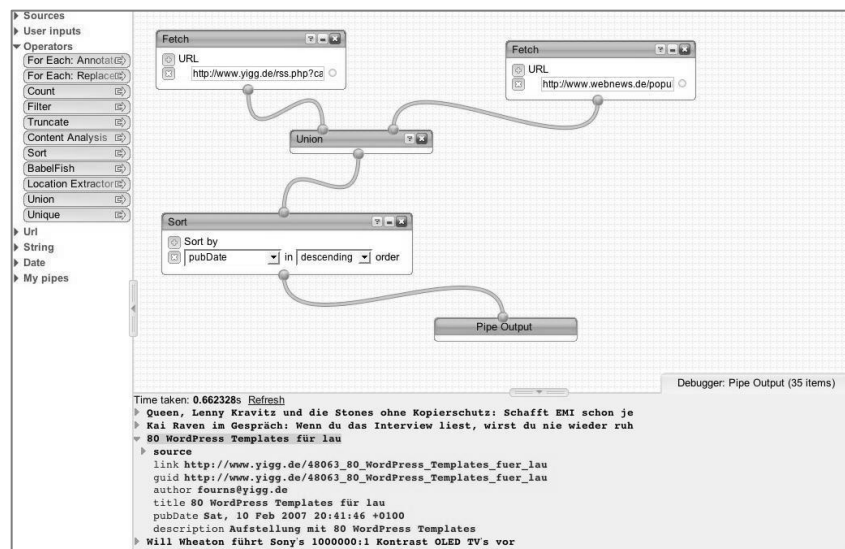


Abbildung 24: Yahoo Pipes

Quelle: (netzwelt.de<sup>31</sup>)

## 2.5.6 Google Now

Der Dienst *Google Now* von Google<sup>32</sup> liefert kontextabhängige Echtzeitinformationen auf Basis von (persönlichen) Informationen, die Google zugänglich sind; außerdem bindet Google eigene Dienste wie beispielsweise *Google Maps* ein. Nutzt der Anwender beispielsweise den Dienst *Google Calendar*, erinnert ihn Google Now an einen Termin mit dem ergänzenden Hinweis, wie lange bei aktueller Verkehrssituation der Anwender für die Fahrt zum Ort des Termins benötigt.

Je mehr Informationen über das Profil des Anwenders existieren, desto mehr Features bietet ihm Google Now. Beispielsweise kann Google Now entsprechend der Interessen eines Anwenders und seines aktuellen Aufenthaltsorts passende Speiselokal- oder Event-Empfehlungen offerieren.

Das Konzept von Google Now ist nicht neu und wird im Rahmen von *Location Based Services* schon über einen längeren Zeitraum untersucht. Der Vorteil von Google bei der Umsetzung des Konzepts besteht in der hohen Anzahl eigener einzubindender Dienste sowie in der Verfügbarkeit einer Vielzahl persönlicher Daten der Dienstnutzer.

<sup>31</sup> Siehe <http://www.netzwelt.de/news/75171-rss-feeds-remixen-yahoo-pipes.html>

<sup>32</sup> Siehe <http://www.google.de>

Im Vergleich zu dem in der vorliegenden Arbeit intendierten Framework besteht eine Gemeinsamkeit in Bezug auf die kontextbasierte „Versorgung“ mit Echtzeitinformationen. Google Plus kann diesbezüglich Anregungen für die Konzeption des Frameworks geben.

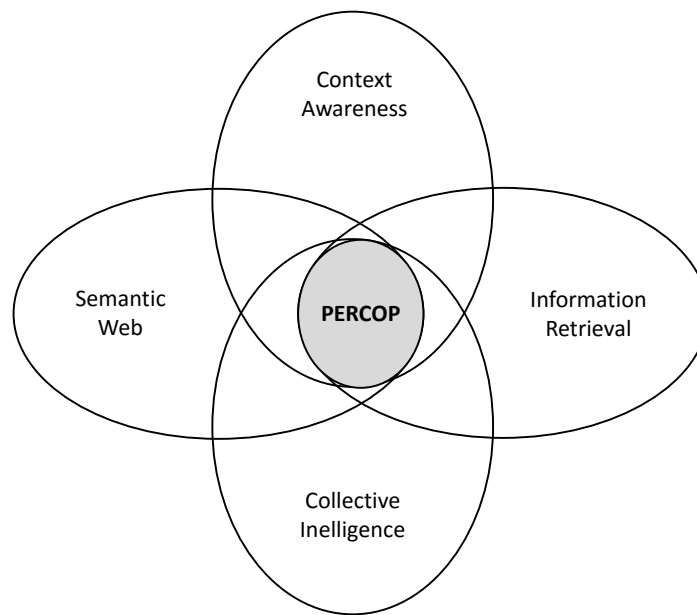
Jedoch unterscheiden sich die Ansätze sowohl in Hinblick auf den Einsatzbereich und der adressierten Zielgruppe als auch im Hinblick auf die Akteure und deren Rollen und Einflussmöglichkeiten. Zudem ist die explizite Beteiligung einer Vielzahl unterschiedlicher Individuen nicht Bestandteil des Google Now-Ansatzes.

## 2.6 Zwischenfazit

Kapitel 1 und Kapitel 2 haben den thematischen Bezugsrahmen der vorliegenden Arbeit aufgespannt und folgende Themengebiete identifiziert: Hauptziel des intendierten Frameworks ist die Beschaffung und Bereitstellung von Informationen zur Unterstützung unterschiedlicher Tätigkeiten, beispielsweise dem Treffen von Entscheidungen, der Erfüllung von Aufgaben oder der Lösung von Problemen. Die Beschaffung von Informationen und deren geeignete Bereitstellung stellen Aufgaben des *Informationsmanagements* dar.

Die Informationsversorgung von einzelnen Personen zur Unterstützung der Durchführung ihrer täglichen Arbeit steht im Fokus des sog. *persönlichen Informationsmanagements* (PIM) als Teilgebiet des Informationsmanagements.

Das intendierte Framework zielt auf eine Echtzeitinformationsversorgung ab. Als Teilgebiet des Informationsmanagements betrachtet die *Informationslogistik* die Informationsflüsse mit dem Ziel, die „richtigen“ Informationen zur „richtigen“ Zeit am „richtigen“ Ort bereitzustellen. Innerhalb des Frameworks sollen die Ermittlung des Informationsbedarfs sowie die Suche nach relevanten Informationen automatisch aus der aktuellen Situation eines Benutzers heraus erfolgen. Die Erkennung und Verwendung situationsbezogener Daten sind Aufgaben *kontextsensitiver Systeme*.



**Abbildung 25: Einordnung von PERCOP**

Abbildung 25 verdeutlicht, wie der Lösungsansatz der vorliegenden Arbeit in relevante Themen-/Forschungsgebiete einzuordnen ist. Er wird als *PERCOP* bezeichnet, eine verkürzte Form der Projektbezeichnung *Personal Copilot*. PERCOP liegt in der Schnittmenge zwischen *Context Awareness*, *Semantic Web*, *Information Retrieval* und *Collective Intelligence*. Aus dem Bereich Semantic Web sind insbesondere die Annotation durch Metadaten, Ontologien und Sprachen von Interesse. Information-Retrieval-Konzepte kommen bei der Suche nach unstrukturierten Daten zum Einsatz. Die Nutzung von kollektiver Intelligenz und Crowdsourcing-Konzepten erlaubt es, Tätigkeiten auf eine große Gruppe von Menschen zu verteilen, um dadurch zum einen eine effektive Verteilung von Aufgaben auf geeignete Stakeholder zu ermöglichen. Zum anderen kann dadurch die Komplexität der einzelnen Aktivitäten reduziert werden.

Die Darstellung verwandter Arbeiten in Abschnitt 2.5 hat gezeigt, dass in verschiedenen Forschungs- und Praxisprojekten Konzepte und Lösungen erarbeitet wurden, die für das intendierte Framework nützlichen Input liefern bzw. zum Teil in das Framework integriert werden können. Insbesondere die Nutzung kollektiver Intelligenz bei der Realisierung einer Echtzeitinformationsversorgung und die Umsetzung in Form eines generischen Frameworks grenzen den Lösungsansatz der vorliegenden Arbeit von den verwandten Arbeiten ab.

Im folgenden zweiten Teil der Arbeit wird das intendierte Framework präsentiert. Des Weiteren wird detailliert auf die im ersten Teil vorgestellten thematischen Bereiche ein-

gegangen, in deren Schnittmenge sich PERCOP einordnet. Dazu zählen neben Informationsbedarf, -beschaffung und -bereitstellung auch die Möglichkeit der Beteiligung einer großen Anzahl („Masse“) Personen, Empfehlungssysteme sowie die Erzeugung von Informationsartefakten. Für jeden Bereich soll im Detail untersucht werden, in welcher Form sich vorhandene Konzepte und Methoden in das intendierte Framework integrieren lassen.

# TEIL II

## Das generische Framework PERCOP

Im zweiten Teil der Arbeit wird das intendierte Framework vorgestellt. Neben der Architektur werden insbesondere die Bereiche Informationsbedarf, -beschaffung und -bereitstellung (Kapitel 4) sowie das Potenzial der Masse (Kapitel 5) diskutiert. Anschließend wird ein Empfehlungssystem präsentiert (Kapitel 6) sowie im Detail auf Informationsquellen und deren semantische Annotation eingegangen (Kapitel 7).

### 3 Das PERCOP-Framework

Die Ausführungen in Teil 1 der Arbeit haben gezeigt, welchen hohen Stellenwert heute relevante Informationen in vielen Situationen in den unterschiedlichsten Domänen besitzen. Die Bereitstellung, die Beschaffung und die Bewertung von Informationen erfordern allerdings komplexe Prozesse. Diese Komplexität kann sich negativ auf die Situation bzw. auf die Handlung oder Entscheidung des Informationsbedarfsträgers auswirken, indem die Suche nach den Informationen lange dauert, die relevanten Informationen aus sehr vielen irrelevanten Informationen herausgefiltert werden müssen oder falsche Informationen „geliefert“ werden.

Für das intendierte Framework sind somit die folgenden Eigenschaften bzgl. der Suche nach relevanten Informationen zu fordern:

- eine kurze bzw. akzeptable Dauer des Suchprozesses,
- ein möglichst geringer manueller Aufwand zur Durchführung des Prozesses, d. h. eine möglichst weitgehende passive/automatisierte Informationssuche und
- eine adäquate Qualität der „gefundenen“ Informationen.

In den folgenden Abschnitten wird das intendierte Framework PERCOP spezifiziert. In Abschnitt 3.1 wird zunächst auf die Zielsetzung des Frameworks eingegangen. Anschließend werden in Form eines Grobkonzepts die Bestandteile und deren semantischen Beziehungen von PERCOP präsentiert (Abschnitt 3.2). Gegenstand von Abschnitt 3.3 ist die Überführung des Grobkonzepts in ein Feinkonzept in Gestalt des Entwurfs der konkreten Framework-Architektur.

#### 3.1 Die Zielsetzung von PERCOP

Im Bereich der Softwareentwicklung werden immer häufiger *Frameworks* eingesetzt, um Entwicklungsaufwand dadurch zu reduzieren, dass grundlegende und häufig erforderliche Funktionalität bereitgestellt wird; diese Funktionalität muss dann nicht mehr immer aufs Neue redundant in den Software-Anwendungen implementiert werden:

*„Sometimes, speaking about software reuse, frameworks are compared to generic components with a high degree of abstraction, but frameworks can probably be best described as the context that defines specific ‘pattern’ of communication and cooperation among software components. Frameworks are tightly coupled to patterns in the way that they could represent the instantiation of a solution to a problem in a defined context.“ (Valerio et al. 1997, S. 4)*

Frameworks sind somit keine eigenständigen Anwendungen; sie stellen vielmehr eine grundlegende Funktionalität bereit und geben dazu eine Grund-Architektur sowie eine Grundstruktur für den Kontrollfluss der jeweils zu realisierenden Anwendung vor.

Ziel der vorliegenden Arbeit ist die Entwicklung eines Frameworks, das die Informationsversorgung von Anwendern bei der Nutzung softwareintensiver Systeme verbessert. *Verbesserung* bedeutet hierbei

- erstens die Reduzierung der Zeitspanne von der Erkenntnis, dass ein bestimmter Informationsbedarf besteht bis zum Erhalt relevanter Informationen,
- zweitens die Reduzierung des manuellen Aufwands für den zugehörigen Suchprozess bzw. umgekehrt ein hoher Automatisierungsgrad und
- drittens die Verbesserung der Qualität der erhaltenen Informationen im Vergleich zu den Suchprozessen, wie sie aktuell durchgeführt werden (vgl. oben).

Darüber hinaus adressiert das Framework den Wissenstransfer der Anwender untereinander. Dieses Charakteristikum ist als prägend für das Framework zu bewerten. Der Begriff *Crowd* wird häufig verwendet, wenn eine große Anzahl voneinander unabhängiger Individuen sich aufgrund gemeinsamer Interessen an einer Aufgabe beteiligen (vgl. auch *Crowd sourcing*, *Crowd funding* und *Crowd simulation*). Daher wird der Begriff auch innerhalb der vorliegenden Arbeit verwendet. Denkbar wäre auch der Begriff *Community*; dieser impliziert jedoch häufig eine kleinere, geschlossene Benutzergruppe, weshalb im Weiteren der Begriff *Crowd* verwendet wird.

Der Wissenstransfer der Anwender untereinander soll für das Framework dahingehend präzisiert und operationalisiert werden, dass unterschiedliche Rollen mit unterschiedlichen Aufgaben für die Mitglieder der *Crowd* identifiziert werden. Die Beteiligung der verschiedenen Rollen kann in unterschiedlichen Bereichen (z. B. bei der Informationserstellung, -anreicherung, Kontextermittlung, -zuordnung etc.) erfolgen (vgl. unten); es ist davon auszugehen, dass der Grad und die Ausprägung der Beteiligung die Qualität der Ergebnisse der Informationssuche beeinflusst.

Die Erreichung der genannten Ziele soll für die entsprechenden Framework-Implementierungen in unterschiedlichen Domänen und für unterschiedliche Technologien übergreifend realisiert werden. In diesem Sinne soll das intendierte Framework „generisch“ sein, d. h., durch seine Nutzung sollen unterschiedlichste Implementierungen die genannten Ziele in der für sie jeweils adäquaten Zielkombination für „ihre“ Anwendungen erreichen können.

## 3.2 Grobentwurf von PERCOP

Das intendierte Framework muss, wie oben analysiert, zur Realisierung seiner Zielsetzung Konzepte, Vorgehensweisen und Verfahren aus verschiedenen Bereichen einbinden, insbesondere aus den Bereichen *Informationsbeschaffung*, *Kontextnutzung* und der *Beteiligung der Masse*. An die Konzepte, Vorgehensweisen und Verfahren dieser Bereiche sind aus Sicht des Frameworks jeweils verschiedene Anforderungen zu stellen; auf diese Anforderungen wird in den folgenden Kapiteln im Detail eingegangen.

Beim Grobentwurf des PERCOP-Frameworks ist zu berücksichtigen, dass die Nutzung des Frameworks für die Erstellung konkreter Implementierungen möglichst einfach sein sollte. Andernfalls könnte nur ein spezieller Anwenderkreis mit entsprechendem Expertenwissen PERCOP nutzen, was wiederum dazu führen könnte, dass das Framework keine signifikante Verbreitung findet<sup>33</sup>, die Transferaufgabe der Arbeit von der Wissenschaft zur Praxis somit nur unzureichend erfüllt würde.

Aufgrund der hohen Bedeutung von Informationen (vgl. noch einmal Kapitel 1) ist für den Bereich der Forschung zu Technologien zur Informationsanreicherung und -suche eine dynamische Entwicklung festzustellen. Damit die Ergebnisse dieser Forschung – insbesondere in Gestalt neuer Verfahren - in die realisierten Framework-Implementierungen integriert werden können, muss das Framework in Bezug auf diese Integration flexibel gestaltet werden.

Für die auf der Basis von PERCOP erstellten Implementierungen sind aus der Zielsetzung des Frameworks folgende Anforderungen abzuleiten:

- *Einfachheit der Nutzung*: Sowohl die Erstellung konkreter Framework-Implementierungen als auch die Nutzung der erstellten Lösung durch die Anwender muss möglichst „einfach“ sein. Eine zu komplizierte Nutzung würde sich insbesondere auch negativ auf die Beteiligung der Crowd auswirken.
- *Transparenz für den Nutzer*: Um die Akzeptanz zu erhöhen, ist eine möglichst hohe Transparenz nötig. Wenn für den Benutzer zu jeder Zeit sichtbar ist, welche Informationen über ihn und sein Verhalten bei der Nutzung des softwareintensiven Systems verwendet werden, führt dies zu mehr Akzeptanz und Vertrauen in die eingesetzte Lösung. Die Art der Nutzung der Informationen über den Benutzer durch das Framework sollte von ihm konfigurierbar sein.

---

<sup>33</sup> Frameworks wie bspw. CORBA (Common Object Request Broker Architecture) zeigen, dass eine zu große Komplexität dazu führen kann, dass vielversprechende Konzepte keine Verwendung finden (vgl. Henning 2008).



- *Unaufdringlichkeit*: Ziel des Frameworks ist die Bereitstellung von Informationen in Echtzeit. Der Benutzer soll dabei jedoch von seiner Tätigkeit nicht abgelenkt werden. Die Präsentation von Informationen soll daher in unaufdringlicher Art und Weise geschehen.
- *Einfachheit der Beteiligung*: Um die Partizipation der Crowd in möglichst vielen Teilen des Konzepts zu ermöglichen, müssen einfache Mechanismen für die verschiedenen Rollen bereitgestellt werden. Komplexe Beteiligungsprozesse können zu geringerer Beteiligungsbereitschaft führen.
- *Flexibilität*: Die zuvor genannte Flexibilität des Frameworks gilt in gleicher Weise für die auf seiner Basis erstellten Implementierungen. Eine hohe Flexibilität führt zu einer größeren Anzahl möglicher Einsatzgebiete und dadurch auch zu einem größeren Kreis an potenziellen Nutzern. Dies wiederum kann sich positiv auf den Grad der Beteiligung auswirken.

Unter Berücksichtigung der Zielsetzung und der genannten Anforderungen wird im Folgenden zunächst ein Grobentwurf der für PERCOP relevanten Objekte und ihrer Beziehungen untereinander vorgestellt. Im nächsten Schritt wird anschließend der Grobentwurf in eine konkrete Framework-Architektur überführt (Abschnitt 3.3).

Der Begriff *Grobentwurf* wird hier im weitesten Sinne verwendet, indem das grundsätzliche Ausgangsszenario und der grundsätzliche Lösungsansatz entwickelt werden soll.

### 3.2.1 Ausgangsszenario

#### *Benutzer*

Zielgruppe des Frameworks sind reale Personen als Benutzer von softwareintensiven Systemen beliebiger Art. Die Benutzer nutzen jeweils „ihr“ System, um bestimmte Aufgaben durchzuführen. Beispielsweise kann ein Benutzer ein Softwareentwickler sein, der eine Entwicklungsumgebung und weitere Werkzeuge für die Softwareentwicklung verwendet, oder ein Grafiker, der ein Bildbearbeitungsprogramm zur Erstellung eines Designs nutzt.

Um eine Aufgabe mit Hilfe eines softwareintensiven Systems durchzuführen, ist unterschiedliches Wissen erforderlich. Dies ist zum einen Wissen über die zu erledigende Aufgabe, d. h., welche Schritte in welcher zeitlich-logischen Reihenfolge abzuarbeiten sind, welche Aktivitäten im Rahmen der einzelnen Schritte in welcher Form durch-

zuführen sind etc. Zum anderen muss der Aufgabenträger als Benutzer des unterstützenden softwareintensiven Systems die Funktionen des Systems kennen und „beherrschen“, die er für die Durchführung der Arbeitsschritte und Aktivitäten einsetzen muss. Insbesondere dann, wenn das softwareintensive System aus einer großen Anzahl unterschiedlicher Softwareanwendungen besteht, ist dieser zweite Wissensbereich als groß und komplex einzustufen.

Hinzu kommt, dass sich das „konkret nutzbare“ Wissen dieses Bereichs dynamisch verändert, da neue Versionen der Softwareanwendungen oder zusätzliche Anwendungen in das softwareintensive System eingebunden werden. Die sich daraus für die Benutzer ergebenden Schwierigkeiten bzw. der zugehörige Informationsbedarf führen weltweit zu vielen Arbeitsplätzen in entsprechenden Help-Desks.

### *Benutzerprofil*

Ein Benutzer besitzt sicherlich in unterschiedlichen Umgebungen unterschiedliche Fähigkeiten: Beispielsweise ist ein Benutzer vielleicht innerhalb einer Umgebung Experte, besitzt er in einer anderen Umgebung nur geringe Kenntnisse. Zudem werden Umgebungen unterschiedlich genutzt; z. B. verwendet ein Web Designer (Rolle) die Anwendung *Adobe Photoshop* (Umgebung) in anderer Art und Weise als ein Bildbearbeiter (Rolle).

Der Informationsbedarf eines Benutzers hängt folglich von den Fähigkeiten ab, die er in einer konkreten Situation besitzt. Im Framework müssen daher für den Informationssuchprozess Informationen über die Benutzer in Form von Benutzerprofilen berücksichtigt werden.

### *Informationsbedarf*

Die bewusste Erkenntnis, dass ein Informationsdefizit besteht, bildet natürlicherweise den Auslöser für die Initiierung eines Suchprozesses nach Informationen zur Beseitigung des Defizits. Allerdings kann auch ein Informationsbedarf bestehen, der nicht bewusst ist.

Bezüglich Suchprozessen haben Untersuchungen gezeigt, dass Menschen häufig keine Suche nach Alternativen mehr durchführen, wenn sie einen Weg gefunden haben, der ihr Problem löst. Dieses Verhalten wird auch als *begrenzte* bzw. *eingeschränkte Rationalität* bezeichnet (vgl. Simon 1959). Es ist somit davon auszugehen, dass „bessere“ Lösungen in einem signifikanten Umfang unentdeckt bleiben; diese Lösungen können ebenfalls als unbewusster Informationsbedarf eingestuft werden.

Entsprechend der formulierten Zielsetzung muss das Framework sowohl den bewussten als auch den unbewussten Informationsbedarf berücksichtigen. Benutzer sollen automatisiert Informationen erhalten, wie sie die Aufgabe, die sie gerade bearbeiten, noch effizienter erfüllen können, und dies unabhängig davon, ob ein bewusstes oder unbewusstes Informationsdefizit besteht.

### *Information*

Relevante Informationen, die zur richtigen Zeit in der richtigen Form/Repräsentation vorliegen, beseitigen Informationsdefizite. Bezüglich der unterschiedlichen Repräsentationen von Informationen im Umfeld von computergestützten Informationssystemen spricht man heute verbreitet von *Artefakten*. Für das Framework ist somit zu fordern, dass es vorsieht, dass - bezogen auf die jeweilige Kombination aus Kontext und Benutzerrolle - die richtigen Informationsartefakte zur Beseitigung des Informationsdefizits bereitgestellt werden. Beispielsweise kann für die Durchführung einer komplexen Funktionsabfolge in einer Anwendung ein Video ein adäquates Informationsartefakt darstellen, das als Screencast die Durchführung der Abfolge demonstriert. In einem anderen Kontext ist möglicherweise ein Blog-Eintrag ein sinnvolles Artefakt.

### *Umgebung*

Die vorliegende Arbeit fokussiert die Nutzung softwareintensiver Systeme. Da aber ein „generisches“ Framework entwickelt werden soll, das für die Realisierung konkreter Frameworks durchaus auch in anderen Bereichen dienen kann, soll im Weiteren der allgemeinere Begriff *Umgebung* statt *softwareintensives System* genutzt werden; damit soll die angestrebte hohe Adaptierbarkeit und Nutzbarkeit von PERCOP betont werden.

### *Adressierte Funktionalität*

Die jeweilige Umgebung ermöglicht es ihren Benutzern, Aufgaben durchzuführen, indem sie eine konkrete Funktionalität zur Verfügung stellt. Beispielsweise stellt eine E-Mail-Anwendung die Funktionen „*Nachricht verfassen*“ und „*E-Mail-Konten verwalten*“ zur Verfügung. Innerhalb einer Projektverwaltungssoftware kann der Anwender z. B. die Funktionen „*Ressourcen verwalten*“ und „*Projektablaufdiagramm erstellen*“ nutzen.

Bezogen auf das intendierte Framework wird die Funktionalität einer Umgebung als Gesamtmenge der dem Benutzer zur Verfügung stehenden Funktionen verstanden. Im Fokus stehen muss aus dieser Gesamtmenge die umgebungsspezifische Kernfunktionalität, da diesbezüglich der größte Nutzen realisiert werden kann.

Der Informationsbedarf des Benutzers bei der Nutzung der Funktionalität einer Umgebung bezieht sich grundsätzlich auf Funktionen, die er jeweils in seiner „aktuellen“ Situation nutzt. Um den Informationssuchprozess adäquat initiieren zu können, muss das Framework somit vorsehen, automatisch die jeweils aktuelle Nutzung der Funktionen durch einen Benutzer zu erkennen.

### 3.2.2 Lösungsansatz

#### *Sensoren*

Zur automatischen Erkennung von Benutzeraktivitäten und von Informationen über die Umgebung müssen sog. *Sensoren* in das Framework eingebunden werden. Sie sollen die Umgebung observieren und ihre ermittelten Daten zur Weiterverarbeitung bereitstellen.

Bei den im Fokus der vorliegenden Arbeit stehenden softwareintensiven Systemen sind Sensoren primär Software-Anwendungen bzw. -Module. Sie sollen beispielsweise erkennen, welche Anwendung aktuell genutzt wird, welche Aktion ein Benutzer jeweils aktuell ausführt und welche Fehlermeldungen die Anwendung eventuell anzeigt. Entwickelt und bereitgestellt werden sollen die Sensoren durch Mitglieder der Crowd.

#### *Regeln*

Die von den Sensoren ermittelten Daten müssen, um die aktuelle Situation identifizieren zu können, miteinander in Verbindung gebracht werden können. Zudem muss spezifiziert werden können, welche Informationen in welcher Situation nützlich sind. Für die entsprechenden logischen Verknüpfungen zwischen Sensordaten, Informationen und Funktionalität müssen Regeln definiert werden können. Entsprechend der im Framework vorgesehenen Trennung zwischen Information auf der einen Seite und Funktionalität auf der anderen Seite sind folgende Fälle zu unterscheiden:

- *Verknüpfung von Sensordaten und Funktionalität*: Durch eine Regel kann definiert werden, bei welchen gemeldeten Sensordaten welche Funktionen in einer Umgebung ausgeführt werden.
- *Verknüpfung von Sensordaten und Information*: Durch eine Regel können direkt relevante Informationsartefakte zu ermittelten Sensordaten zugeordnet werden.
- *Verknüpfung von Funktionalität und Informationen*: Durch eine Regel können Informationen direkt entsprechenden Funktionen zugeordnet werden.

Die Trennung ermöglicht eine hohe Flexibilität sowohl in Bezug auf die „Zuordnungsmächtigkeit“ als auch für die Beteiligung der Crowd. Beispielsweise kann eine Person eine

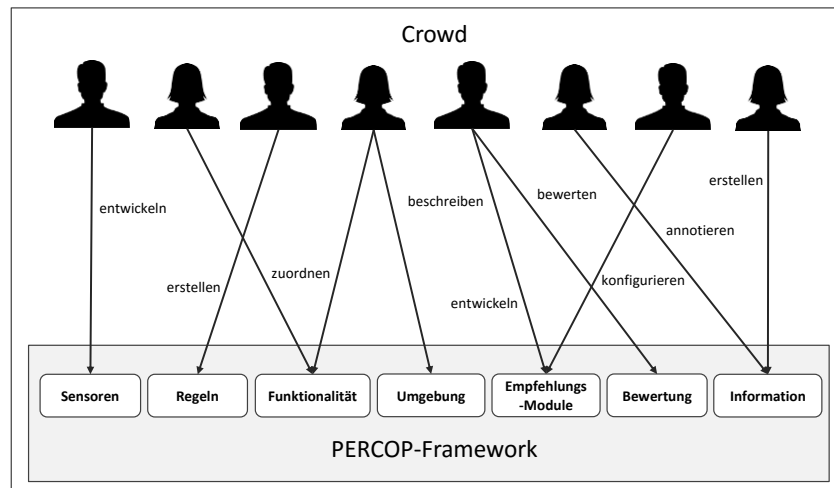
Regel definieren, die eine bestimmte Aktivität eines Benutzers einer bestimmten Funktion zuordnet (z. B. wird das Öffnen eines bestimmten Fensters innerhalb einer E-Mail-Anwendung durch Sensoren erkannt und im Rahmen einer Regel der Funktion „*Neue Nachricht verfassen*“ zugeordnet). Davon zunächst einmal unabhängig/entkoppelt kann ein anderer Benutzer ein Informationsartefakt direkt einer konkreten Funktionalität zuordnen (z. B. „Zeige bei Funktion ‚*Neue Nachricht verfassen*‘ die Information ‚XYZ‘ an“). Durch Auswertung der Regeln durch ein entsprechendes situationsgetriebenes Regel-Verarbeitungssystem kann auf Basis der Regeln jeweils automatisch ermittelt werden, welche Informationen für einen Benutzer bei der Ausführung bestimmter Funktionen nützlich sein können.

### *Crowd*

Im Kontext des Frameworks bezeichnet die Crowd die Gesamtheit aller Stakeholder, die an folgenden Aktivitäten beteiligt sind:

- Umgebungen auswählen,
- Regeln erstellen,
- Funktionen zuordnen,
- Informationen zuordnen,
- Informationsartefakte erstellen,
- Informationsartefakte semantisch anreichern,
- Sensoren entwickeln,
- Empfehlungsmodule (Recommender) entwickeln,
- Empfehlungskonfigurationen (Recommender-Konfigurationen) definieren und
- Informationsvorschläge bewerten.

Jede Aktivität soll entsprechend der angesprochenen Trennung grundsätzlich von einer anderen Person durchgeführt werden können, dies auch dann, wenn die Aktivitäten in einem Zusammenhang stehen, beispielsweise eine erstellte Regel sich auf die Daten von bestimmten Sensoren bezieht (vgl. Abbildung 26). Dies ermöglicht die Beteiligung von Personen entsprechend ihrer Rolle, ihrer Fähigkeiten und Interessen, ohne dass zwischen ihnen irgendeine Beziehung bestehen muss. Beispielsweise muss eine Person, die eine Regel definiert, nicht wissen, welche Person den Sensor entwickelt hat, dessen Daten innerhalb der Regel verwendet werden.



**Abbildung 26: Einflussmöglichkeiten der Crowd**

### *Recommender*

Um unterschiedliche Informationsquellen nach konkreten Informationsartefakten durchsuchen zu können, sind entsprechende Werkzeuge notwendig. In das PERCOP-Framework sollen daher sog. Recommender integriert werden. Als einen Recommender bezeichnet man eine Empfehlungskomponente, die in der Lage ist, auf Basis von Eingabeparametern in einer bestimmten Informationsbasis nach entsprechenden Informationsartefakten zu suchen, diese zu filtern und bereitzustellen.

In Hinblick auf die Entwicklung und Kombination der Recommender, sollten diese feingranularer Art und lose gekoppelt sein. Dadurch können einzelne Recommender sinnvoll wiederverwendet und zu neuen Recommendern kombiniert werden. Beispielsweise soll ein Recommender in einem bestimmten Expertensystem nach relevanten Einträgen suchen können, ein anderer Recommender nutzt z. B. eine Suchmaschine für das Auffinden von Informationsartefakten.

Die Recommender sollen von Mitgliedern der Crowd unabhängig voneinander entwickelt und bereitgestellt werden. Mehrere Recommender sollen anschließend zu einer Recommender-Konfiguration kombiniert werden können.

### *Recommender-Konfiguration*

Das Konzept der Recommender-Konfiguration soll eine maßgeschneiderte Zusammenstellung nützlicher Recommender für eine konkrete Domäne bzw. Umgebung ermöglichen. Eine Recommender-Konfiguration soll als Kombination mehrerer Recommender selbst wieder eine Empfehlungskomponente auf höherer Ebene bilden; Input und Output sollen deshalb grundsätzlich dem In-/Output eines einzelnen Recommenders

entsprechen. Analog zu den Services auf den verschiedenen Ebenen einer SOA soll dadurch eine hohe Wiederverwendung der einzelnen „feingranularen“ Recommender ermöglicht werden.

Des Weiteren sollen im Rahmen der Erstellung und Verwaltung von Recommender-Konfigurationen die Fähigkeiten der unterschiedlichen Crowd-Mitglieder bestmöglich genutzt werden. Während die „feingranularen“ Recommender-Module durch Softwareentwickler programmiert werden, sind bei der Erstellung einer Recommender-Konfiguration grundsätzlich keine Programmierkenntnisse nötig; vielmehr werden dazu Fachwissen und Kenntnisse über geeignete Informationsquellen für konkrete Umgebungen und Situationen benötigt.

Recommender-Konfigurationen sollen ebenfalls durch die Crowd domänenspezifisch erstellt, einzeln bewertet und optimiert werden. Bei den jeweiligen Tätigkeiten können unterschiedliche Verfahren aus dem Bereich der Empfehlungssysteme Verwendung finden, beispielsweise kollaborative, hybride, oder inhaltsbasierte Verfahren.

In Kapitel 6 wird das Konzept des Empfehlungssystems innerhalb von PERCOP weiter konkretisiert.

### **3.2.3 Zusammenfassende Darstellung des Grobentwurfs**

Abbildung 27 fasst den Entwurf von PERCOP zusammen, indem die wesentlichen Objekte des Frameworks und deren semantischen Beziehungen zueinander visualisiert werden. Die Darstellung adressiert zunächst nur das Gesamtverständnis des Frameworks, bereitet die detaillierte Betrachtung der einzelnen Teilbereiche sowie die Konzeption und Erstellung der Architektur vor, indem jeweils der Bezug der einzelnen Konzepte zum Gesamtkonzept spezifiziert wird.

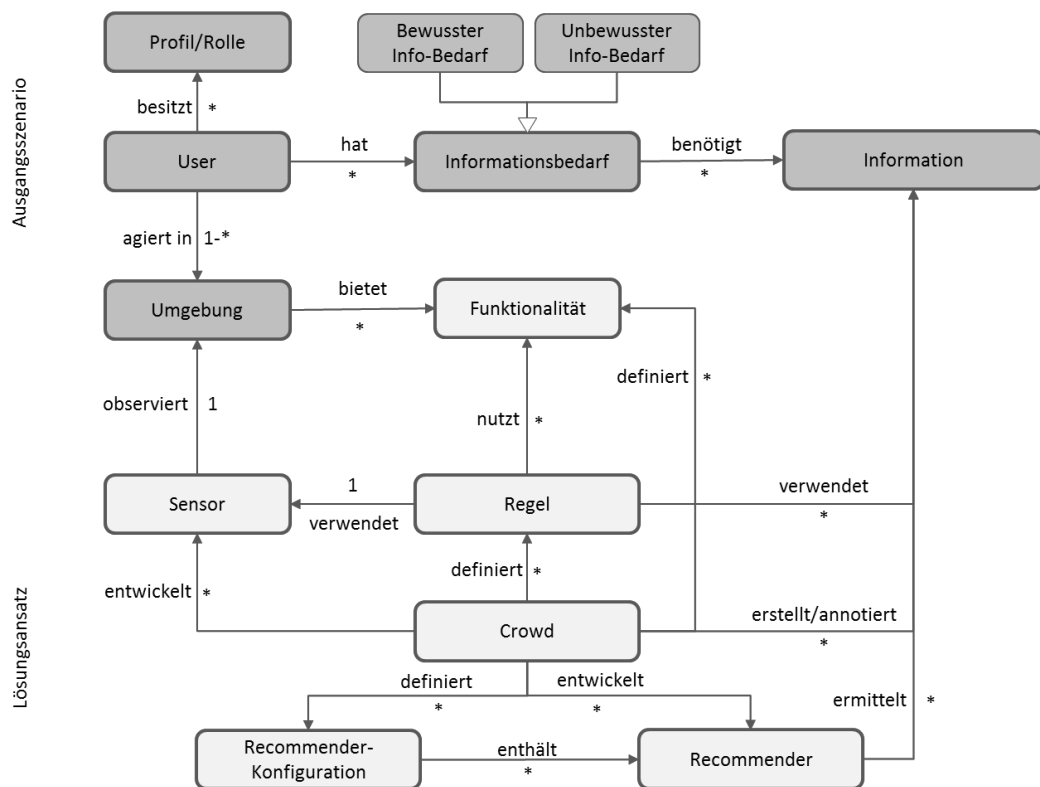


Abbildung 27: Grobentwurf des Frameworks

Im Folgenden werden die Zusammenhänge erläutert: Benutzer agieren innerhalb von Umgebungen und besitzen abhängig von ihrer Tätigkeit innerhalb der jeweiligen Umgebungen ein Profil, welches Benutzereigenschaften wie beispielsweise den Status (z. B. Anfänger, Experte), die Sprache oder die bevorzugten Informationstypen (z. B. Dokument, Video) enthält.

Umgebungen stellen ihren Benutzern verschiedene Funktionen bereit. Während ihrer Interaktion mit der Umgebung bzw. der Nutzung der Funktionen haben Benutzer jedoch häufig Bedarf nach Informationen, um die jeweilige Aufgaben mit den jeweiligen Funktionen durchführen zu können oder Probleme bei der Durchführung lösen zu können.

Der Bedarf nach Informationen kann einerseits bewusst, andererseits unbewusst sein. „Gestillt“ wird der Bedarf durch relevante Informationen. Sensoren überwachen dazu die Interaktionen in der Umgebung und liefern Daten zur Identifizierung der jeweiligen aktuellen Situation. Regeln verknüpfen Daten und Informationen von Sensoren mit Funktionalitäten und Informationen.

Eine zentrale Rolle in dem Framework spielt die Crowd. Dies wird insbesondere deutlich durch die hohe Anzahl der Beziehungen zu anderen Bestandteilen des Frameworks, wie



sie in Abbildung 27 dargestellt werden: Einzelne Mitglieder der Crowd entwickeln Sensoren, definieren Regeln, spezifizieren Funktionen und erzeugen Informationen unterschiedlicher Art. Des Weiteren entwickeln sie Recommender und/oder konfigurieren diese zu Recommender-Konfigurationen.

Wesentliches Merkmal des beschriebenen Entwurfs ist die Unabhängigkeit der unterschiedlichen Aktivitäten durch Mitglieder der Crowd. Dadurch können die spezifischen Fähigkeiten, Kenntnisse und Interessen der einzelnen Crowd-Mitglieder bestmöglich genutzt werden.

Bezugnehmend auf den zuvor beschriebenen Grobentwurf können zur bestmöglichen Nutzung aller Bestandteile des Frameworks folgende Voraussetzungen identifiziert werden.

- Die Benutzer sind bereit, ihr Verhalten innerhalb der genutzten IT-Umgebung in Hinblick auf die Informationsbeschaffung (durch Sensoren) erfassen zu lassen.
- Umgebungsspezifische Benutzerinformationen werden in Form von Profilen hinterlegt.
- Autoren von Informationsartefakten oder andere Personen reichern die Artefakte mit Metadaten an bzw. es erfolgt eine automatische Ergänzung durch Metadaten.
- Personen sind bereit, sich zu beteiligen (z. B. bei der Erstellung von Sensoren, Regeln, Zuordnungen oder Recommendern).
- Benutzer sind bereit, erhaltene Informationen in Hinblick auf ihre Relevanz zu bewerten.

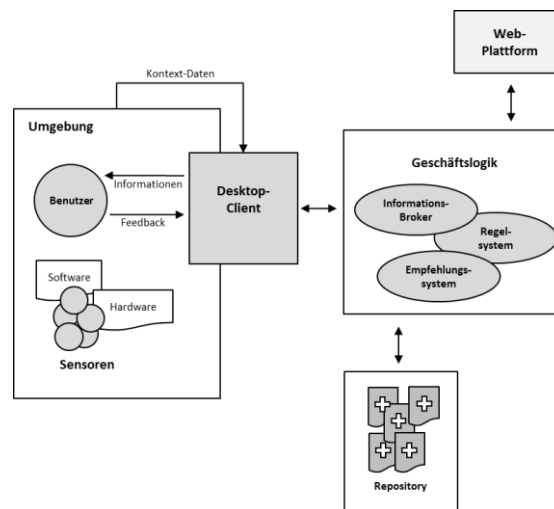
Die genannten Prämissen sind in ihrer Gesamtheit sicherlich nicht zwingend notwendig für die Nutzbarkeit des Frameworks; vielmehr ist eine konkrete Implementierung auf Basis des Frameworks auch einsetzbar, wenn einzelne Voraussetzungen nicht gegeben sind. Wurden beispielsweise Informationen nicht mit kontextspezifischen Metadaten angereichert, können die Informationen trotzdem gefunden werden; allerdings sinkt in diesem Fall die entsprechende Wahrscheinlichkeit. Werden Profile der Benutzer nicht abgelegt, sinkt möglicherweise die Qualität der ermittelten Informationen hinsichtlich ihrer Relevanz, da die Profilinformationen nicht als Parameter für Suchanfragen genutzt werden können.

### **3.3 Architektur und Konzeption von PERCOP**

Der im vorausgehenden Abschnitt präsentierte Grobentwurf fokussiert das Verständnis zum Betrachtungsgegenstand und den zugehörigen relevanten Begriffen, Elementen und

semantischen Zusammenhängen. Im Folgenden soll der Grobentwurf in eine konkrete Framework-Architektur übertragen werden. Abbildung 28 visualisiert voraussichtlich die Komponenten und Beziehungen der Architektur; im Folgenden werden diese Komponenten und Beziehungen sukzessiv entworfen und erläutert.

Zuvor sollen jedoch – ebenfalls voraussichtlich – zwei einfache Beispiele präsentiert werden, um einen Eindruck über die Komponenten und ihren Beziehungen zu verschaffen.



**Abbildung 28: PERCOP-Architektur**

Folgendes Beispiel demonstriert den Einsatz von Sensoren innerhalb von PERCOP: In Form eines Software-Moduls wird auf der Plattform ein virtueller Sensor bereitgestellt. Des Weiteren wird definiert, welche Daten der Sensor ermitteln kann und in welchen Wertebereichen die Daten liegen. Ist ein Sensor beispielsweise in der Lage, die Fensterbeschriftung des aktuell aktiven Fensters zu ermitteln, kann dadurch im nächsten Schritt bestimmt werden, welche Anwendung der Benutzer in diesem Moment verwendet und welche Funktionalität innerhalb der Anwendung gerade ausgeführt wird. Gleichzeitig kann beispielsweise ein zweiter Sensor Informationen über den aktuellen Benutzer ermitteln.

In einem konkreten Szenario können diese Sensoren und Werte innerhalb der Regeln beispielsweise wie folgt genutzt werden: Ein Mitarbeiter erstellt z. B. ein Informationsdokument zu Bewertungskriterien (*Bewertungskriterien.pdf*). Diese Informationen sollen alle Mitarbeiter erhalten, wenn sie an Bewertungsdokumenten arbeiten (und zwar nur dann). Dazu definiert der Mitarbeiter unter Verwendung der PERCOP-Plattform die folgende Regel:

*WENN Sensor1.AktuelleAnwendung = "Microsoft Word" UND*  
*Sensor1.AktuellesDokument = "\*Bewertung\*.doc" UND*  
*Sensor2.BenutzerGruppe = "Mitarbeiter"*  
*DANN InformationPush= "Bewertungskriterien.pdf"*

Angewendet würde diese Regel dafür sorgen, dass jeder Benutzer der Gruppe Mitarbeiter automatisch das definierte Dokument angezeigt bekommt, wenn er ein Microsoft-Word-Dokument bearbeitet, welches im Namen den Begriff „Bewertung“ enthält.

Ein weiteres Beispiel ist die automatische Ermittlung der Komplexität einer Softwarearchitektur innerhalb einer Software-Entwicklungsumgebung und der davon abhängigen Ermittlung von relevanten Informationen. Im Beispiel wird Fuzzylogik verwendet, worauf in Abschnitt 3.3.3 detailliert eingegangen.

1. Zunächst wird ein Sensor spezifiziert und entwickelt, z. B. mit Hilfe eines Plug-Ins für die Entwicklungsumgebung Visual Studio. Der Sensor ist in der Lage, unterschiedliche Daten aus der Entwicklungsumgebung heraus zu ermitteln.
2. Anschließend werden Attribute definiert, welche von dem Sensor ermittelt werden können und für das Szenario relevant sind, z. B. *Anzahl der Klassen, Anzahl der Abhängigkeiten, Anzahl der Fremdkomponenten* oder *Metriken*.
3. Danach werden die Wertebereiche der Attribute festgelegt, beispielsweise absolute Werte oder Intervalle.
4. Für jedes Attribut werden verschiedene linguistische Werte definiert, z. B. *einfach, komplex, sehr einfach, sehr komplex, nicht sehr komplex, mehr oder weniger einfach*.
5. Für jeden linguistischen Wert wird die Fuzzy-Menge festgelegt, beispielsweise die Dreieck-, Trapez oder Gauß-Funktion inkl. Nutzung von Operatoren und kontextfreier Grammatik.
6. Regeln werden definiert.

Im Rahmen von PERCOP können somit unter Verwendung der Fuzzylogik sensorübergreifende Regeln definiert werden, die einfach zu formulieren sind und sprachliche Elemente integrieren, wie beispielsweise:

- WENN CPU-Lüfter-Drehzahl *niedrig* UND CPU-Temperatur *hoch*  
DANN Lüftersteuerung ist *defekt*
- WENN Lüftersteuerung *defekt*  
DANN Meldung\_Information\_A
- WENN Bildschirmauflösung *sehr klein* UND Umgebungsname ist *Photoshop*  
DANN Meldung\_Hinweis\_Auflösung

- WENN Umgebungsdomäne ist *mail* UND Aktion ist *neue\_mail*  
UND MailAnhang ist *false*  
UND TextContent enthält *Schlüsselwörter\_für\_Anhang*  
DANN: Meldung\_Möglicherweise\_Anhang\_vergessen

### 3.3.1 Umgebung

Entsprechend des Grobentwurfs umfasst die „Umgebung“ alle Objekte, mit denen ein Benutzer jeweils interagiert. Wie bereits mehrfach angesprochen, stehen im Fokus der Arbeit Softwareanwendungen einschließlich der darunterliegenden Basis, insbesondere dem Betriebssystem.

Um das Verhalten und die Aktionen eines Benutzers erfassen zu können, wird eine Möglichkeit zur Beobachtung einer Umgebung benötigt. Im Hinblick auf entsprechende Zugriffsoptionen können drei Typen von Umgebungen identifiziert werden (vgl. auch Beul und Eicker 2009 und Beul und Eicker 2012):

- White-Box-Umgebung,
- Black-Box-Umgebung und
- Gray-Box-Umgebungen.

*White-Box-Umgebungen* ermöglichen den vollen Zugriff auf kontextrelevante Daten; dies ist aus Sicht des entwickelten Ansatzes als „ideal“ einzustufen. Insbesondere neu zu entwickelnde Softwareanwendungen können bzw. sollten wegen der entsprechenden Vorteile als White-Box-Umgebungen realisiert werden; denn dadurch besteht die Möglichkeit, Sensoren zur Übermittlung von kontextrelevanten Informationen direkt in die Anwendungslogik einzubinden. Beispielsweise können mit Hilfe entsprechender Komponenten von allen Ebenen der Softwareanwendung (Geschäftslogik, User-Interface, Datenbank-Zugriff, Web-Service-Aufruf etc.) kontextrelevante Daten vergleichsweise aufwandsarm bereitgestellt werden.

*Black-Box-Umgebungen* gestatten keinen direkten Zugriff auf die Funktionalität des betrachteten Systems. Unglücklicherweise sind ein Großteil der existierenden (und wohl-möglich auch der zukünftigen) Anwendungen Black-Box-Umgebungen, erlauben somit keinen Zugriff auf Daten/Zustände der internen Systemlogik. Für solche Anwendungen wird somit ein alternativer Ansatz benötigt, der es ermöglicht, Benutzeraktivitäten und Anwendungszustände „von außerhalb“ zu erfassen. Eine Möglichkeit, den entsprechenden Zugriff zu realisieren, bieten virtuelle Sensoren; auf diese wird in Abschnitt 3.3.3 näher eingegangen.

Eine Kombination aus White-Box- und Black-Box-Umgebung wird als *Gray-Box-Umgebung* bezeichnet. Diese Form der Umgebung ermöglicht partiellen Zugriff auf ihre Bestandteile. Dieser partielle Zugriff ergibt sich daraus, dass heutzutage eine Vielzahl von Anwendungen auf erweiterbare Frameworks wie beispielsweise der *Eclipse Rich Client Platform (RCP)* basiert. Solche Frameworks ermöglichen die Integration von Plug-Ins, um die Funktionalität der auf ihnen basierenden Anwendungen zu erweitern. Dadurch wird ein Zugriff auf Anwendungsfunktionen sowie auf Zustandsinformationen ermöglicht, ohne dass Änderungen an der Anwendungslogik durchgeführt werden müssen.

Um die Validität des entwickelten Ansatzes überprüfen zu können, werden in den Fallbeispielen Szenarios für alle drei Typen von Umgebungen praktisch umgesetzt: Fallbeispiel 1 repräsentiert eine GrayBox-Umgebung, bei der die Integration durch Plug-Ins realisiert wird (Abschnitt 9.2). Eine White-Box-Umgebung liegt in Fallbeispiel 2 vor, die Integration erfolgt direkt innerhalb der konkreten Anwendungslogik (Abschnitt 9.3). In Fallbeispiel 3 wird eine Black-Box-Umgebung durch eine Anwendung repräsentiert, die keinen Zugriff auf Programmfunktionalität ermöglicht und somit durch externe Sensoren überwacht werden muss (Abschnitt 9.4).

### 3.3.2 PERCOP-Client

Der PERCOP-Client bildet die Schnittstelle zwischen dem Benutzer und den Informationen, die möglicherweise in Bezug auf die aktuelle Situation des Benutzers nützlich sind. Der PERCOP-Client hat im Wesentlichen zwei Aufgaben: Zum einen stellt er dem Benutzer grafisch aufbereitet die ermittelten Informationen bereit, zum anderen nimmt er Feedback des Benutzers im Hinblick auf die Relevanz der dargestellten Informationen entgegen; dieses Feedback gibt der Client an die PERCOP-Geschäftslogik weiter.

Aus den allgemeinen Anforderungen an die Framework-Implementierung (vgl. Abschnitt 3.2) sind aufgrund der Aufgaben der Benutzerschnittstelle folgende Anforderungen an den Desktop-Client abzuleiten:

- *Konfigurierbar*: Aus der allgemeinen Anforderung nach Transparenz ergibt sich die Anforderung an den Client, dass der Benutzer die Anwendung an seine Bedürfnisse anpassen kann. Eine Anpassung kann beispielsweise durch Änderung der Darstellungsart geschehen, oder durch die Möglichkeit zur Einflussnahme, welche Profil- oder Umgebungsinformationen zur Informationssuche verwendet werden.
- *Unaufdringlich*: Aus der allgemeinen Anforderung nach Unaufdringlichkeit ergibt sich die Anforderung an den Client, dass er den Benutzer nicht von seiner Tätigkeit

ablenken soll, sondern „unauffällig“ auf möglicherweise relevante Informationen hinweist.

- *Intuitiv bedienbar*: Aus der allgemeinen Anforderung nach Einfachheit ergibt sich die Anforderung an den Client, dass jeder Benutzer seine wesentlichen Funktionen auch ohne Einarbeitung oder Anpassungen verwenden kann.
- *Plattformunabhängig*: Aus der allgemeinen Anforderung nach „generischer“ Umsetzung ergibt sich die Anforderung an den Client, dass er für alle relevanten Betriebssysteme entwickelt bzw. auf allen Plattformen genutzt werden kann.
- *Echtzeitfähig*: Aus der allgemeinen Anforderung nach Relevanz ergibt sich die Anforderung an den Client, dass er die ermittelten Informationen ohne signifikante Verzögerungen bzw. „innerhalb“ der jeweils der zugehörigen Benutzertätigkeit präsentiert.

Abbildung 29 zeigt einen ersten Entwurf der Oberfläche des Clients in Gestalt einer Layout-Skizze. Die (schmale) spaltenartige Grundstruktur soll es dem Benutzer ermöglichen, den Client während der Arbeit mit anderen Softwareanwendungen am Rand des Bildschirms zu platzieren. Diese Darstellungsform wird bereits bei vielen Messaging-Anwendungen eingesetzt und ermöglicht die gleichzeitige Nutzung/Sichtbarkeit beider Anwendungen.

Im oberen Bereich des Desktop-Clients (1) werden Informationen über den aktuellen Benutzer und die Daten des aktuell ermittelten Kontexts angezeigt. Der Anwender sieht zu jeder Zeit, welche Daten zur Informationssuche genutzt werden. Unterhalb des Bereichs befinden sich Filtermöglichkeiten (2). Beispielsweise kann der Benutzer Sprache und Informationstyp selektieren, was direkt innerhalb der Informationssuche durch das Empfehlungssystem berücksichtigt wird.

Der Hauptbereich (3) stellt die Liste der ermittelten Informationsartefakte dar. Der Benutzer erhält in der Übersicht Informationen über Inhalt, Art und Herkunft der Information und kann diese selektieren (4). Des Weiteren ist es möglich, direkt innerhalb der Liste eine Information als hilfreich oder nicht hilfreich bezogen auf den Benutzer und seinen aktuellen Kontext zu bewerten.

Die einzelnen Umgebungen werden als Reiter (sog. *Tabs*) dargestellt. Wechselt der Benutzer die Umgebung oder die Anwendung, wird automatisch auf den entsprechenden Reiter gewechselt.

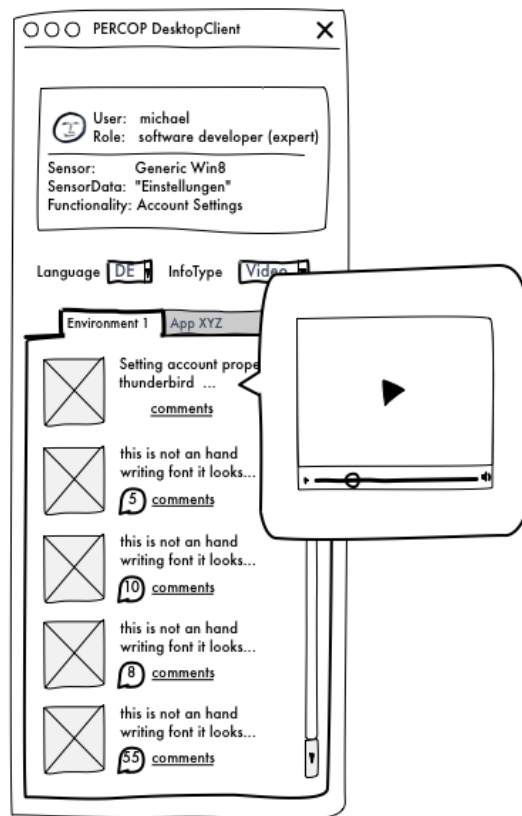


Abbildung 29: Layout-Skizze des PERCOP-Clients

### 3.3.3 Sensoren

#### *Sensoren innerhalb von Softwaresystemen*

Für den Begriff Sensor existiert keine einheitliche Definition. Aus technischer Sicht repräsentieren Sensoren die „Schnittstelle zwischen dem elektronischen Regelsystem auf der einen Seite und der Umwelt, dem Prozeß, dem Verfahren oder der Maschine auf der anderen Seite“ (Hauptmann 1991, S. 11). Eine allgemeine Beschreibung liefert Schaumburg:

*„Ein SENSOR ist das primäre Element in einer Meßkette, das eine variable Eingangsgröße in ein geeignetes Meßsignal umsetzt.“ (Schaumburg 1992, S. 1)*

Im Gegensatz zu den Ausführungen von Schiessle (vgl. Schiessle 1992) beschränkt sich diese Beschreibung nicht auf physikalische Sensoren; sie wird daher im Weiteren als Definition eines Sensors zugrunde gelegt.

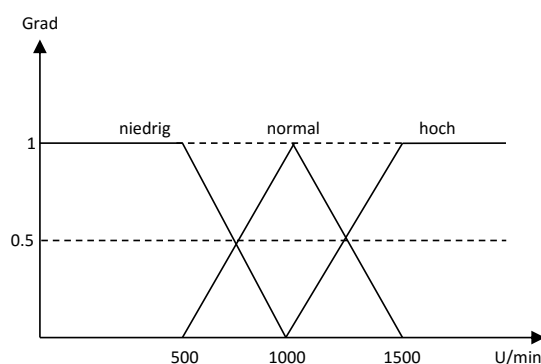
Sensoren können in drei Gruppen klassifiziert werden (vgl. Indulska und Sutton 2003 und Baldauf et al. 2007, S. 4):

- *Physikalische Sensoren*: Hardware-Sensoren, die in der Lage sind, physikalische Daten aus der Umwelt zu ermitteln, beispielsweise Luftdruck, Helligkeit, Temperatur oder Ort.
- *Virtuelle Sensoren*: Software-Sensoren, die Daten aus einer Software-Anwendung ermitteln können, beispielsweise Mausposition, Tastatureingaben oder die Systemzeit.
- *Logische Sensoren*: Sensoren als Kombination aus physikalischen und virtuellen Sensoren, die mit Hilfe weiterer Informationen komplexere „Aufgaben“ lösen können.

### Fuzzylogik

1965 von Zadeh entwickelt, ermöglicht die Fuzzylogik, aus sprachlich formulierten Sätzen, Begriffen oder Regeln, die Erzeugung von mathematischen Beschreibungen, die maschinell verarbeitbar sind (vgl. Zadeh 1965). Somit handelt es sich bei der Fuzzylogik um ein wissensbasiertes System, d. h., es fließt vorhandenes Wissen zum Prozess in die Verarbeitung ein.

Für Sensoren und Regeln reicht häufig die zweiwertige Boolesche Logik nicht aus, um einen Sachverhalt adäquat zu beschreiben, da in vielen Fällen „fließende“ Übergänge zwischen Werten existieren. Eine Temperatur kann beispielsweise mit *kalt*, *warm* oder *heiß* bestimmt werden. Werden diesen Werten konkrete Grad-Zahlen zugeordnet, kann eine gemessene Temperatur einen dieser drei Werte ergeben. Jedoch ist möglicherweise auch relevant, ob etwas beispielsweise *sehr kalt* oder *ein wenig heiß* ist. Für die Modellierung solcher *Unschärfen* wurde die Theorie der Fuzzylogik entwickelt.



**Abbildung 30: Fuzzylogik – Linguistische Variablen**

Abbildung 30 zeigt die Kurve einer Fuzzylogik-Funktion, die die Fuzzy-Variable *Geschwindigkeit* (U/min) mit den linguistischen Termen *niedrig*, *normal* und *hoch* definiert.



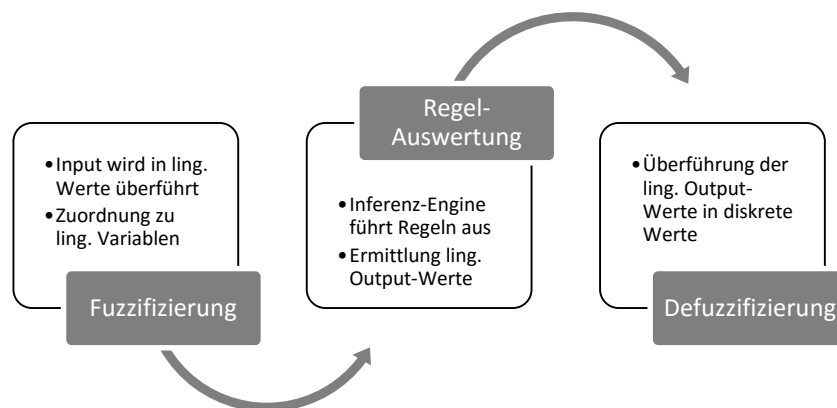
Während bei der Booleschen Logik ein Objekt nur entweder Teil einer Gruppe sein kann (1) oder nicht Teil der Gruppe ist (0), sind bei der Fuzzylogik mit Hilfe der sog. *Mitgliedschaftsfunktionen* (sog. *Member Functions*) auch alle Werte zwischen 0 und 1 möglich.

„Ein Fuzzy Set  $A$  in  $X$  ist durch eine Zugehörigkeitsfunktion  $f_A(x)$  charakterisiert, die jedem Punkt in  $X$  eine reelle Zahl im Intervall  $[0,1]$  mit dem Wert  $f_A(x)$  zuordnet, die den „Grad der Zugehörigkeit“ von  $x$  zu  $A$  darstellt.“ (Zadeh 1965, S. 339) (ins Deutsche übersetzt).

Mit Hilfe der Fuzzylogik kann somit beispielsweise die Geschwindigkeit innerhalb von Abbildung 30 zu 30% niedrig und 70% normal sein, d. h., der Grad der Mitgliedschaft wäre 0,3 (niedrig) und 0,7 (normal). Anders formuliert liegt die Geschwindigkeit zwischen niedrig und normal, ist aber „*eher normal*“.

Es existieren unterschiedliche Arten von Mitgliedschaftsfunktionen, beispielsweise Trapez-, Dreiecks oder Gauß-Funktionen. Dadurch ermöglicht die Fuzzylogik zum einen die korrekte mathematische Abbildung logischer Zwischenstufen, zum anderen können Objekte - mit jeweils unterschiedlichen Graden - gleichzeitig Mitglied in mehreren Fuzzy-Sets sein.

Innerhalb von Fuzzylogik-Systemen werden häufig die drei Abschnitte Fuzzifizierung (*Fuzzyfication*), Inferenz/Regelbasis (*Rule Evaluation*) und Defuzzifizierung (*Defuzzyfication*) durchlaufen (vgl. Abbildung 31 und Chen und Pham 2001, 145ff.).



**Abbildung 31: Phasen eines Fuzzylogik-Systems**

Quelle: In Anlehnung an (Chen und Pham 2001, S. 148)

Bei der Fuzzifizierung werden zunächst die gemessenen Werte (scharfe Eingangswerte) in einen Vektor von Zugehörigkeitsgraden der linguistischen Variablen überführt. Anschließend werden die Regeln ausgewertet (*Rule Evaluation*). Dabei nutzt eine Fuzzy-Inferenz-Engine die Regelbasis und ermittelt dadurch die linguistischen Output-Werte. Schließlich

wird bei der Defuzzifizierung aus der erhaltenen unscharfen Wertemenge wieder eine scharfe Ausgangsgröße ermittelt.

### *Sensoren innerhalb von PERCOP*

Innerhalb von PERCOP müssen je nach Typ der Umgebung Sensoren unterschiedlicher Art eingesetzt werden. Beispielsweise kann ein Sensor in einer Black-Box-Umgebung Statusinformationen des Betriebssystems erfassen. In einer Gray-Box-Umgebung ermöglichen Sensoren in Gestalt von Plug-In-Modulen z. B. die Ermittlung von nützlichen Daten zu bestimmten Anwendungen. Handelt es sich um eine White-Box-Umgebung, können Sensoren Daten direkt aus der Anwendung heraus bereitstellen.

Zu unterscheiden sind *aktive* und *passive* Sensoren: Aktive Sensoren senden ihre ermittelten Daten selbständig, passive Sensoren stellen ihre Daten erst auf Anfrage bereit.

Die Art der Übertragung durch die Sensoren darf im Framework nicht festgelegt werden, um der Anforderung nach generischer Umsetzung zu entsprechen. Für die Integration der Crowd und der Echtzeit-Informationsbeschaffung ist jedoch eine einfache, reduzierte und standardisierte Darstellung für die Übertragung zu fordern und festzulegen, um die Anforderung *Relevanz* zu erfüllen.

Als Darstellungsform sind sog. *Key-Value-Paare* sehr gut geeignet, weil lediglich zwei Werte übertragen werden, ein eindeutiger Schlüssel und dessen Wert. Dazu sind über die Crowd-Plattform (siehe Abschnitt 3.3.7) jeder Umgebung und deren Funktionalität, jedem Informationsartefakt sowie jedem Sensor und dessen Daten eindeutige Bezeichner zuzuordnen. Ein Sensor muss lediglich seine Sensor-Daten-Id (key) und den gemessenen Wert (value) übertragen; die Plattform kann dann ausgehend von diesen beiden Werten auf die Bedeutung des Messwerts schließen und relevante Regeln zur Ausführung bringen.

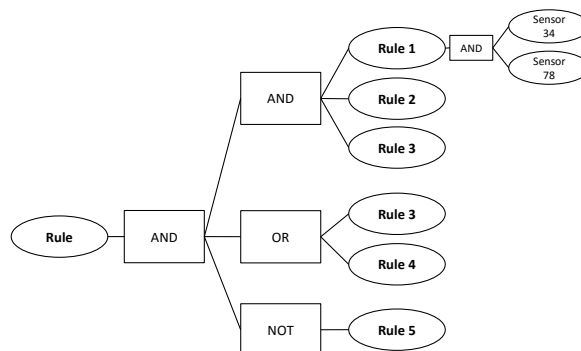
Für das Framework sind insbes. virtuelle Sensoren relevant, zum einen, weil Werte physikalischer Sensoren für die Ziele des Frameworks überwiegend eine untergeordnete Rolle spielen. Zum anderen werden innerhalb von Software-Anwendungssystemen Werte physikalischer Sensoren in der Regel durch Treiber in virtuelle Sensoren umgewandelt. Beispielsweise wird aus den GPS-Koordinaten eines (physikalischen) GPS-Sensors von einem virtuellen Sensor die konkrete Adresse berechnet und zur Verfügung gestellt.

Die Verwendung linguistischer Werte fördert zudem den Aspekt der Beteiligung durch die Crowd, da die Regeln unkompliziert und natürlich-sprachlich formuliert werden können.

### 3.3.4 Regelsystem

Regeln verbinden Sensordaten und damit implizit Kontexte und Situationen mit relevanten Informationen. Dabei können Regeln Sensordaten direkt mit Informationen verknüpfen oder indirekt durch die Verknüpfung mit Funktionen (die wiederum mit Informationen verknüpft werden können). Des Weiteren besteht die Möglichkeit, ohne die Verwendung von Sensoren, Informationen zu Funktionen zuzuordnen.

In Hinblick auf die Beteiligung durch die Crowd wurde Wert daraufgelegt, ein Regelsystem zu entwickeln, welches auf der einen Seite einfach in der Anwendung, auf der anderen Seite jedoch mächtig genug ist, um komplexe Kontext-Verknüpfungen zu ermöglichen. Daher wurde eine Kombination aus Boolescher Algebra (mit den Operationen *UND*, *ODER*, *NICHT*) und Fuzzylogik-Konzepten entwickelt. Diese Kombination ermöglicht auch die Definition komplexer Regeln, die trotz der Komplexität auch für Menschen lesbar sind. Abbildung 32 zeigt eine konkrete Regel und verdeutlicht, dass jede Regel aus einer Kombination von (Teil-)Regeln und Sensoren besteht.



**Abbildung 32: PERCOP Regel-System**

Die Definition eines Sensors beinhaltet eine eindeutige *Id*, *Version*, *Wert*, *Datentyp* und gültige *Operatoren*. Des Weiteren können Fuzzylogik-Information innerhalb der Regeldefinition verwendet werden. Abbildung 33 zeigt eine XML-Darstellung der in Abbildung 32 dargestellten Regel. Einer der Sensoren (Id 78) beinhaltet linguistische Fuzzylogik-Ausdrücke, um Werte mit Intervallen verwenden zu können (Operator="equals", Wert ="high").

```
<Rule id="199">
  <AND>
    <AND>
      <Rule id="1">
        <AND>
          <Sensor id="34" operator="equals" value="high" />
          <Sensor id="78" operator="greaterthan" value="50" />
        </AND>
      <Rule id="2" />
      <Rule id="3" />
    </AND>
    <OR>
      <Rule id="3" />
      <Rule id="4" />
    </OR>
    <NOT>
      <Rule id="5" />
    </NOT>
  </AND>
</Rule>
```

**Abbildung 33: XML-Darstellung von Regeln**

Das Regelsystem ist losgelöst vom Sensorsystem. Dadurch wird eine Trennung der Prozesse und beteiligten Rollen bei der Sensor-Erstellung und der Regel-Erfassung ermöglicht. Dadurch können technisch versierte Personen bzw. Experten im Bereich Sensorprogrammierung und Anwendungsintegration Sensoren entwickeln und veröffentlichen. Andere Personen, die beispielsweise über fachliches Know-how in Gestalt von fundiertem Fach- und Prozesswissen verfügen, benötigen nur Information über die Sensoren und deren Daten und können diese bei der Definition von Regeln verwenden.

### 3.3.5 Direkte Zuordnungen als Ergänzung zum Regelsystem

Das Framework sieht zusätzlich zu dem Regel- und Recommender-System die Möglichkeit der direkten Zuordnung (sog. *Mapping*) vor: Informationsartefakte können direkt entsprechenden Funktionen oder Regeln zugeordnet werden. Es kann zwischen *expliziten* und *impliziten* Zuordnungen unterschieden werden, wobei beide sowohl die aktive als auch die passive Informationsbeschaffung unterstützen.

Explizite Zuordnungen weisen ein Informationsartefakt konkret einer Regel zu. Dieser Zuordnungstyp wird im Kontext des Frameworks mit einem hohen Relevanzgrad in Verbindung gebracht, da dort keine Information-Retrieval-Funktionalität genutzt wird. Die

Zuordnung erfolgt zwar subjektiv, wird jedoch durch das Crowd-Konzept einer ständigen Validierung unterzogen.

Implizite Zuordnungen weisen Informationen einer Funktionalität zu. Wird über das Regelsystem erkannt, dass ein Benutzer aktuell eine Funktionalität nutzt, wird im zweiten Schritt das zugeordnete Informationsartefakt bereitgestellt. Die impliziten Zuordnungen sind sehr einfach – insbesondere durch die Ersteller der Informationsartefakte - zu erstellen. Es muss lediglich die Umgebung und die entsprechende Funktionalität (z. B. *Microsoft Outlook, Kontoeinstellungen bearbeiten*) ausgewählt und durch einen Verweis auf das Informationsartefakt mit diesem verknüpft werden.

### 3.3.6 Information Broker

Aufgabe des Information Broker ist die Koordination der Informationsbeschaffung. Dazu erhält die Komponente Informationen aus der Umgebung wie beispielsweise die aktuelle Rolle des Benutzers, die gerade ausgeführte Aktion oder der Zustand der Umgebung. Diese Informationen bereitet der Information Broker auf und leitet sie an die jeweilige Recommender-Konfiguration weiter. Das Ergebnis leitet der Information Broker in Form einer Liste zurück an den PERCOP-Client.

Des Weiteren nimmt der Information Broker Feedback des Benutzers bzgl. der erhaltenen Informationsartefakte entgegen. Diese Bewertungen und nutzt er dann im Rahmen der Informationsbeschaffung zur Identifizierung relevanter Informationen.

### 3.3.7 Crowd-Plattform

Die Crowd-Plattform bildet die Basis für die Verwaltung aller Tätigkeiten der Crowd. Dazu zählen die Erstellung und Bearbeitung von:

- Umgebungen, Funktionalitäten und Informationen,
- Recommender-Modulen und -Konfigurationen,
- Sensoren und Regeln sowie
- Zuordnungen.

Insbesondere die Aspekte der Beteiligungsmotivation gilt es innerhalb der Crowd-Plattform zu unterstützen bzw. zu fördern. Beispielsweise ermöglichen Kollaborationssysteme eine effiziente Zusammenarbeit zwischen verteilt arbeitenden Individuen. Diese Systeme bieten sich insbesondere im Bereich der Entwicklung von Sensoren und Recommender-Modulen an.

In Kombination mit den Bewertungen der Benutzer über den PERCOP-Client können auch Belohnungssysteme umgesetzt werden, um Anreize zu schaffen, Regeln oder Zuordnungen zu definieren. Dazu können auch aus dem Bereich der Expertensysteme und Communities (z. B. *Experts Exchange*<sup>34</sup> oder *Stack Overflow*<sup>35</sup>) bekannte Verfahren Verwendung finden; diese arbeiten häufig mit Punktesystemen.

### 3.3.8 Informations-Repository

Das Informations-Repository repräsentiert die Datenbasis für alle zu persistierenden Daten innerhalb von PERCOP. Neben den Geschäftsobjekten können innerhalb des Repository auch Informationsartefakte oder Referenzen auf diese mit Metadaten angereichert werden. Drei unterschiedliche Repository-Typen können identifiziert werden (vgl. auch Beul und Eicker 2009, S. 3):

- lokales Informations-Repository,
- globales Informations-Repository und
- hybrides Informations-Repository.

Ein Benutzer oder eine Organisation können eigene lokale Repositories einsetzen, um eine interne Datenbasis oder Wissensdatenbank aufzubauen. Auch interne Dokumenten- und Content-Managementsysteme können als lokale Repositories Verwendung finden.

Ein globales Repository steht einem grundsätzlich offenen Benutzerkreis zur Verfügung. Darunter fallen beispielsweise über das Internet frei zugängliche Expertensysteme und Foren.

Auch die Kombination aus lokalen und globalen Repositories soll durch das Framework unterstützt werden. Über die Plattform sollen dazu Daten zwischen den Repositories transferiert werden können: dazu können die Recommender-Module und –Konfigurationen eingesetzt werden.

Inhalte der Repositories können neben den Informationsartefakten auch Zuordnungsinformationen sowie Ontologien oder Annotations-Schemata sein, worauf in Kapitel 7 detailliert eingegangen wird.

---

<sup>34</sup> Siehe <http://experts-exchange.com>

<sup>35</sup> Siehe <http://stackoverflow.com>

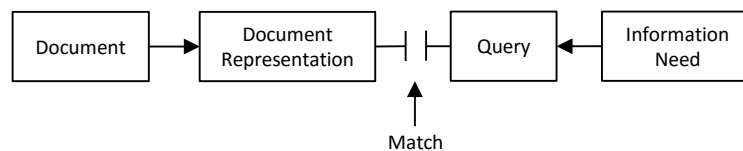
## 4 Informationsbedarf, -beschaffung und -bereitstellung

Das vierte Kapitel gibt zunächst einen Überblick über den State of the Art des Information Retrieval (IR) (Abschnitt 4.1). Anschließend werden die Informationsbedarfsanalyse und die Merkmale des Benutzerverhaltens erläutert (Abschnitt 4.2 und 4.3), um dann die analysierten Aspekte des Informationsbedarfs, der -beschaffung und der -bereitstellung auf das PERCOP-Framework zu übertragen (Abschnitt 4.4).

### 4.1 Grundlagen des Information Retrieval

Der Bedarf nach Information Retrieval und auch die Forschung in diesem Bereich begründen sich im Wesentlichen durch zwei Sachverhalte, zum einen durch die wachsende Anzahl verfügbarer Informationen und zum anderen durch die Bedeutung von relevanten Informationen im Moment des Bedarfs.

Das „klassische“ Information Retrieval befasst sich mit dem Auffinden von (Text-)Dokumenten bezüglich einer konkreten Anfrage zur Deckung eines Informationsbedarfs. Abbildung 34 visualisiert das entsprechende IR-Modell.



**Abbildung 34: Klassisches Information Retrieval Modell**

Quelle: (Bates 1989, S. 408)

Auf der einen Seite befinden sich das Dokument und seine entsprechende Repräsentation, auf der anderen Seite existiert Informationsbedarf, der unter Verwendung einer Anfragesprache (engl. *query language*) in Form einer formalisierten Anfrage (engl. *query*) formuliert werden muss. Zwischen der Anfrage und der Dokumentrepräsentation findet das sog. *Matching*, also der Abgleich von Anfrage und Dokument statt.

Abbildung 35 zeigt eine detailliertere Darstellung eines IR-Modells; insbesondere bezieht das Modell einen Bewertungsprozess ein, der Rücksprünge und Veränderungen der Objektrepräsentation und der Anfrage ermöglicht.

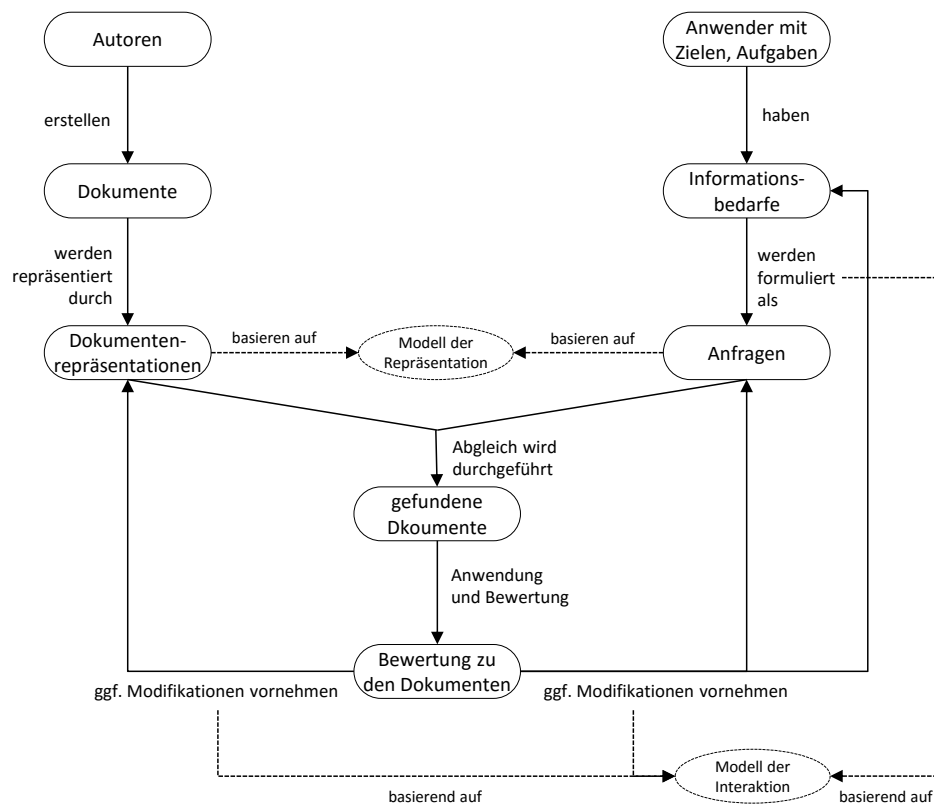


Abbildung 35: Ein allgemeines Modell zum Information Retrieval

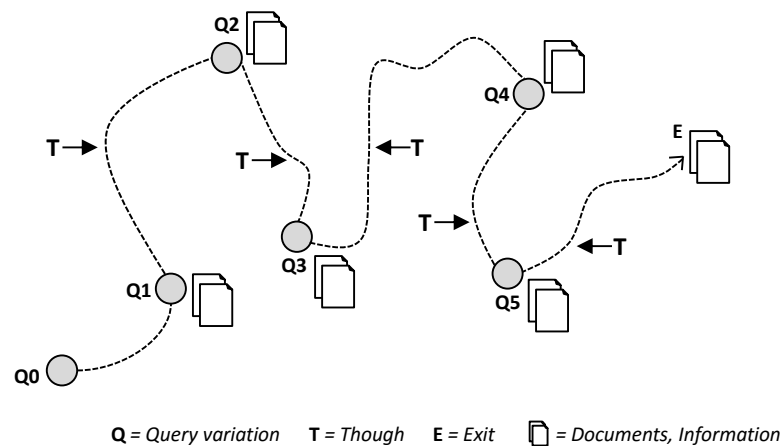
Quelle: (Kuopka 2004, S. 9)

Fuhr charakterisiert ein IR-System als ein „*Informationssystem für vage Anfragen und unsicheres Wissen*“ (Fuhr 1992). Seine Definition stellt darauf ab, dass die Antworten auf Anfragen nicht von vorneherein fest definiert sind; vielmehr repräsentiert eine Anfrage den Wunsch des jeweiligen Benutzers nach einer konkreten Information, wobei die Anfrage in unterschiedlicher Weise formuliert werden kann.

„Unsicherheit“ bezieht Fuhr auf die Eigenschaft des persistierten Wissens sowie auf die Repräsentation der Semantik: Zum einen ist die Überführung eines realen Objekts in die entsprechende Objektrepräsentation in verschiedenen (insbes. technisch-wissenschaftlichen) Anwendungen durch Unsicherheit geprägt, da häufig keine vollständige bzw. nur eine ungenaue Erhebung aller Eigenschaften erfolgen kann. Zum anderen ist vielfach auch die Ableitung der Semantik aus der Objektrepräsentation mit Unsicherheit behaftet (vgl. Fuhr 2010, S. 8).

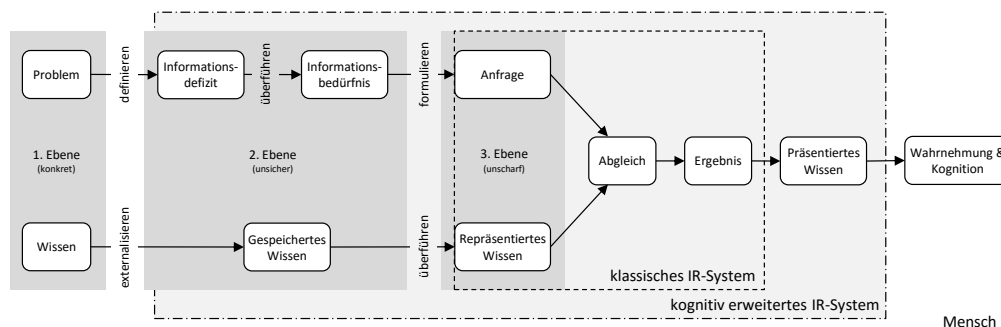
Das Prinzip der fortlaufenden Veränderung der Anfrage zur Ermittlung des Ergebnisses wird als *Berrypicking*-Ansatz bezeichnet (siehe Abbildung 36). Nach jedem Teilergebnis wird die Anfrage mit Hilfe der Informationen, die der Benutzer bisher erhalten hat, verändert, bis schließlich das gesuchte Objekt ermittelt wurde.



**Abbildung 36: Berry picking**

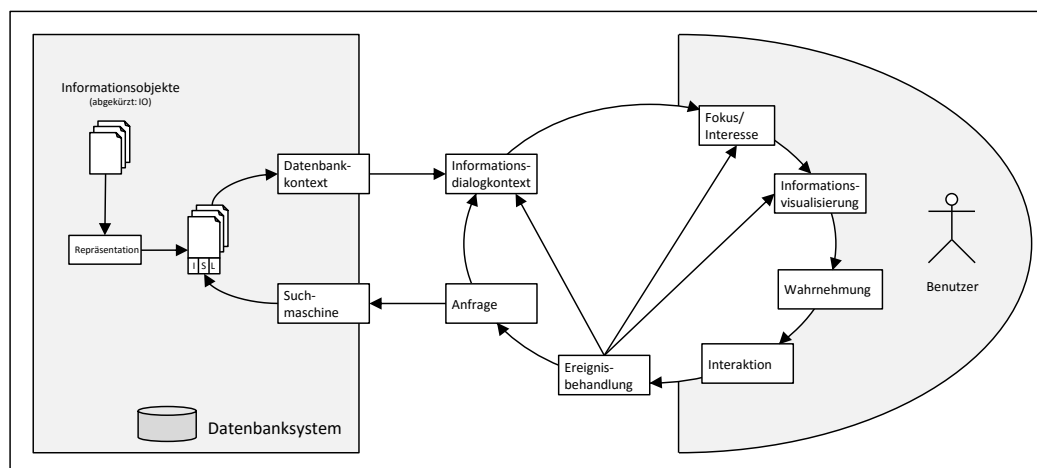
Quelle: (Bates 1989, S. 410)

Eine Erweiterung des konzeptionellen Modells liefert Fuhr durch die Integration kognitiver Eigenschaften (vgl. Fuhr 1992, 5ff.). Sein erweitertes Modell (siehe Abbildung 37) fügt dem klassischen Modell eine zweite Ebene hinzu, die zum einen dem Problem und der daraus resultierenden Anfrage die Definition des Informationsdefizits und die Überführung in ein Informationsbedürfnis hinzufügt. Zum anderen wird vorhandenes Wissen zunächst in gespeichertes Wissen externalisiert und anschließend in repräsentiertes Wissen überführt. Darüber hinaus wird das Ergebnis klassischer Information-Retrieval-Systeme zunächst in repräsentiertes Wissen überführt und dann schließlich vom Menschen wahrgenommen.

**Abbildung 37: Kognitiv erweitertes Modell für Information Retrieval**

Quelle: (Landwich et al. 2007, S. 2)

Landwich et al. integrieren das konzeptionelle Modell und das interaktive Visualisierungsmodell von Hemmje (vgl. Hemmje 1999) in ein konzeptionelles Rahmenmodell für ein interaktives Information-Retrieval-System (vgl. Landwich et al. 2007). Abbildung 38 zeigt das Modell.



**Abbildung 38: Rahmenmodell für visuell direkt-manipulatives Information Retrieval**

Quelle: (Landwich et al. 2007, S. 3)

Ziel des Rahmenmodells ist die Abbildung der Interaktion mit dem Benutzer in Form von Zyklen zur systematischen Unterstützung der Informationssuche. Dadurch wird insbesondere eine Reduzierung der Anfrage-Unschärfe angestrebt. Im Kontext von PERCOP bietet es sich an, diese Eigenschaften im Hinblick auf eine Automatisierung zu berücksichtigen (siehe Abschnitt 4.4).

### 4.1.1 Der Begriff der Relevanz

Die zentrale Aufgabe eines IR-Systems beinhaltet die Ermittlung von Informationen in Form konkreter Informationsartefakte. Die Qualität der identifizierten Informationen wird häufig über den Grad der Relevanz bewertet.

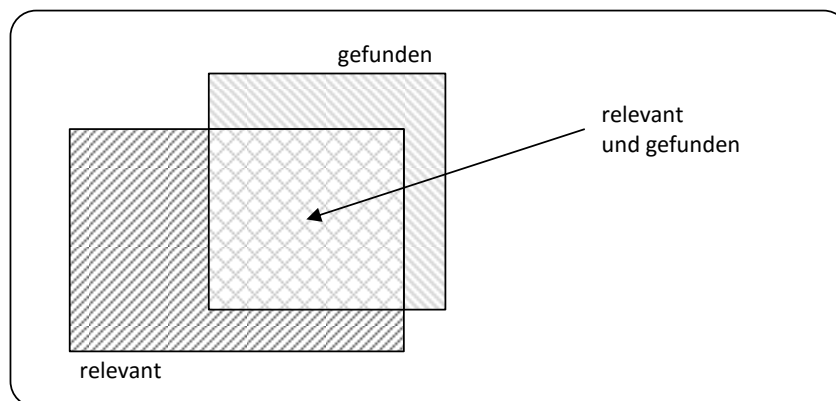
Zur Bewertung der Relevanz existiert in der Literatur eine Vielzahl Veröffentlichungen: Mizzaro identifiziert unterschiedliche Arten der Relevanz in Form von Beziehungen zwischen einer Information, einem Dokument oder einem Surrogat auf der einen Seite, und dem Problem, dem Informationsbedürfnis, dem Anliegen und der Anfrage auf der anderen Seite. In der Kombination ergeben sich dadurch zwölf unterschiedliche Arten der Relevanz; die Dimension Zeit erhöht die Komplexität der Kombinationen noch weiter (vgl. Mizzaro 1997, S. 812). Fuhr unterscheidet im Kontext von IR-Systemen vier Arten der Relevanz (vgl. Fuhr 2010, S. 10):

- Situative Relevanz,
- Pertinenz,
- Objektive Relevanz und
- Systemrelevanz.

Die *situative Relevanz* beschreibt die theoretische Nützlichkeit eines Dokuments in Beziehung zu seiner Aufgabe. Die subjektive Nützlichkeit, die von dem Anwender empfunden wird, wird als *Pertinenz* bezeichnet. Die Pertinenz bezieht sich nicht auf die Anfrage, sondern auf das Informationsbedürfnis des Benutzers. Wird die Relevanz von neutralen Beobachtern beurteilt, wird sie als *objektiv* bezeichnet. Die durch ein System bewertete Relevanz gegenüber einer Anfrage wird als *Systemrelevanz* bezeichnet.

Zusammenfassend kann festgestellt werden, dass eine Information generell relevant sein kann, aber für einen konkreten Benutzer eventuell keine Pertinenz besitzt, da dieser beispielsweise diese Information schon kennt. Für den Informationsbedarf und das Informationsbedürfnis bedeutet dieser Zusammenhang, dass sich die Relevanz auf den objektiven Informationsbedarf bezieht, während die Pertinenz das subjektive Informationsbedürfnis fokussiert.

Ein „ideales“ IR-System würde alle relevanten Informationen und nur diese finden. Häufig sind jedoch zum einen nicht alle relevanten Informationen in der Ergebnismenge enthalten und zum anderen sind auch Informationen enthalten, die nicht relevant sind. Ein Optimierungsansatz kann darin bestehen, die Anzahl der enthaltenen irrelevanten Informationen zu minimieren. Abbildung 39 visualisiert diese Zusammenhänge.



**Abbildung 39: Mengen der relevanten und gefundenen Dokumente**

Quelle: In Anlehnung an (Fuhr 2010, S. 13)

### 4.1.2 Retrieval-Modelle

Für IR-Modelle werden im Wesentlichen drei Ansätze unterschieden, das *Boolesche Modell*, das *Vektorraummodell* und das *probabilistische Modell*: Das *Boolesche Modell* ermittelt Artefakte durch die Formulierung von Suchanfragen unter Verwendung der Booleschen Logik. Es verlangt nach einer exakten Übereinstimmung zwischen der Anfrage und den dazu passenden Dokumenten (*exact match*) (vgl. Lewandowski 2005, S. 80). Diese

Eigenschaft wird in der Literatur häufig kritisiert, da relevante Dokument unentdeckt bleiben, wenn sie der Anfrage nicht vollständig entsprechen. Zudem werden möglicherweise Dokumente gefunden, die zwar der Anfrage entsprechen, jedoch keine Relevanz besitzen. Dieser Fall tritt beispielsweise dann auf, wenn zwar alle Bestandteile (z. B. Wörter) einer Anfrage innerhalb eines Dokumentes vorkommen, diese jedoch nicht in dem relevanten Kontext zueinander stehen (vgl. Lewandowski 2005, S. 80).

Mit Hilfe des *Vektorraummodells* können nicht nur exakte Übereinstimmungen gefunden werden, sondern auch Ähnlichkeiten zwischen der Anfrage und einem Dokument. Denn es wird ein Vektorraum aufgespannt, in dem die Dimensionen durch die Terme und die Dokumente durch Vektoren abgebildet werden. Dadurch ist es möglich, den Ähnlichkeitsgrad festzulegen bzw. eine Sortierung entsprechend dieses Ähnlichkeitsgrads durchzuführen.

Auch das *probabilistische Modell* fokussiert die Ähnlichkeiten zwischen Dokumenten, indem insbesondere die Häufigkeit des Vorkommens der Suchbegriffe innerhalb des Dokumentes betrachtet wird.

Tabelle 2 fasst die wesentlichen Eigenschaften der drei Modelle zusammen.

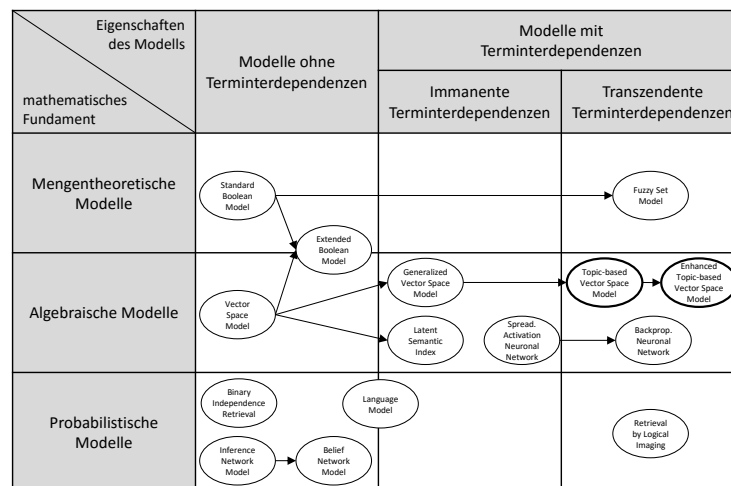
	Boolesches Modell	Vektorraummodell	Probabilistisches Modell
Boolesche Verknüpfungen	Ja		
Gewichtung		Ja	Ja
Ranking		Ja	Ja
Kriterium der Übereinstimmung	Vorhandensein der Begriffe	Vektordistanz	Häufigkeit der Begriffe
Alleinstellungsmerkmal		Relevance Feedback	

**Tabelle 2: Eigenschaften der drei IR-Modelle**

Quelle: (Chu 2003, S. 112)

Insbesondere durch die Weiterentwicklung des Internets haben sich zusätzliche Retrieval-Modelle etabliert, beispielsweise das Netzwerk-Modell, das Nutzermodell und linktypologische Verfahren. Linktypologische Verfahren sind u. a. in der Lage, Ranking-faktoren zu bestimmen, um die Relevanz von Informationen zu ermitteln<sup>36</sup>.

Abbildung 40 ordnet zusammenfassend die Modelle des Information Retrieval im Hinblick auf ihre Eigenschaften und ihrem mathematischen Fundament ein.



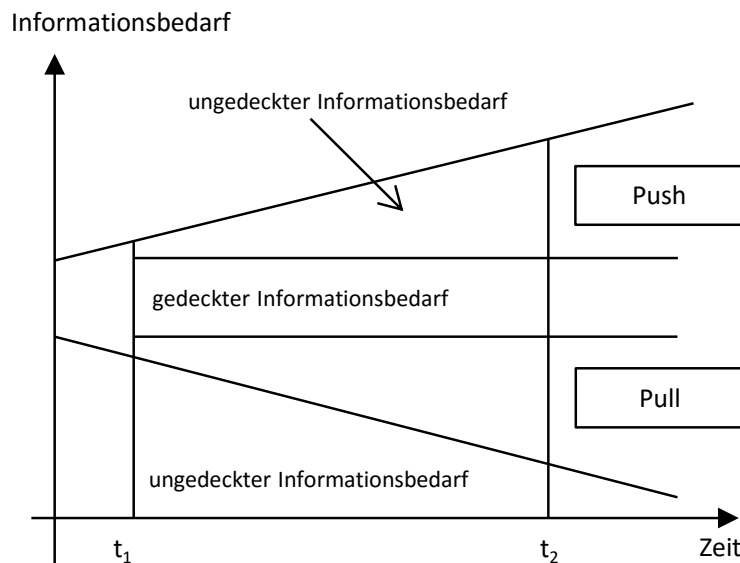
**Abbildung 40: Eine Übersicht über gängige Modelle zur Repräsentation von natürlich-sprachlichen Dokumenten**

Quelle: (Kuropka 2004, S. 44)

### 4.1.3 Pull und Push als Strategien des Information Retrieval

Das Information Retrieval unterscheidet zwei Strategien, wie die Informationen zu den Informationsbedarfsträgern (Benutzern) gelangen können (vgl. Stock 2007, S. 58): Entweder fordert der Benutzer (aktiv) die Informationen aus einem System an (Pull), oder das System überträgt die Informationen („ungefragt“) an den Benutzer (Push). Abbildung 41 setzt die beiden Strategien in Bezug zum Informationsbedarf.

<sup>36</sup> Detaillierte Ausführungen finden sich bei (Lewandowski 2005).



**Abbildung 41: Deckung des Informationsbedarfs**

Quelle: (Stock 2007, S. 59)

Beide Verfahren besitzen jeweils Vor- und Nachteile bzgl. der Anwendung innerhalb eines IR-Systems: Bei einer Pull-Strategie besteht insbesondere der Nachteil, dass der Benutzer selbständig aktiv werden muss, was von ihm die Erkenntnis eines bestehenden Informationsbedarfs erfordert.

Demgegenüber kann die Push-Strategie einen Benutzer mit Informationen versorgen, ohne dass er aktiv werden muss. Diese automatische Informationsversorgung kann aber den Benutzer durch eine häufige unaufgeforderte Lieferung von Informationen von seiner Arbeit ablenken, was negativ zu bewerten ist.

#### 4.1.4 Web Information Retrieval

Suchmaschinen des WWW (World Wide Web, Kurzform: Web) haben sich zum elementaren Recherche-Instrument sowohl für den Unternehmens- als auch für den Privatbereich entwickelt. Die Abgrenzung von WEB-IR-Systemen zu klassischen IR-Systemen soll hier nicht im Detail diskutiert werden; vielmehr sei auf eine entsprechende umfassende Diskussion von Lewandowski verwiesen (Lewandowski 2005). In diesem Abschnitt folgt lediglich eine kurze Übersicht über die wesentlichen Eigenschaften von Web-IR-Systemen und ihren Unterschieden zu klassischen IR-Systemen.

Web-IR-Systeme unterscheiden sich von klassischen IR-Systemen insbesondere aufgrund folgender Eigenschaften bzgl. der vorhandenen Datenmenge und der darin enthaltenen Dokumente (Lewandowski 2005, S. 75):

- Exakte Daten-/Dokumentenmenge ist nicht bekannt/bestimmbar,
- Die Dokumente liegen in unterschiedlichen Sprachen und Formaten vor,
- die Dokumente besitzen eine unterschiedliche Länge und Granularität,
- die Dokumente sind häufig nicht oder nicht einheitlich strukturiert,
- es existieren Dubletten,
- die Dokumente sind miteinander verknüpft,
- die Qualität der Dokumente variiert und
- unterschiedliche Nutzertypen (Laie, Profi) nutzen dasselbe IR-System.

Aufgrund der Eigenschaften sind zusätzliche Konzepte für die Informationsermittlung nötig. Beispielsweise werden mit sog. *Rankingverfahren* die Ergebnislisten so sortiert, dass die Objekte mit der größten Relevanz am Anfang der Ergebnisliste erscheinen und Objekte mit der geringsten Relevanz am Ende (vgl. Lewandowski 2005, S. 89). Der sog. *Page Rank* weist dazu jedem Dokument einen statischen Wert zu. Dieser Wert wird bei den verschiedenen Suchmaschinen jeweils durch unterschiedliche Algorithmen berechnet; er ist grundsätzlich unabhängig von den Suchanfragen, soll vielmehr die Qualität eines Treffers/gefundenen Dokuments einschätzen. Wesentliche Parameter für den Page Rank sind heute häufig die Verknüpfungen von Dokumenten untereinander.

#### 4.1.5 Evaluationskriterien für IR-Systeme

Für die Bewertung von IR-Systemen können unterschiedliche Kriterien herangezogen werden. Dazu wird in der Literatur unterschieden zwischen der Bewertung der *Effektivität* eines IR-Systems und seiner *Effizienz*: Effektivität bezieht sich auf „die Fähigkeit des Systems, dem Nutzer die Informationen nachzuweisen, die er sucht“ (Salton und MacGill 1987, S. 172). Effizienz bezieht sich auf die Kosten und die Zeit, „die zur Ausführung bestimmter Systemoperationen erforderlich sind“ (Salton und MacGill 1987, S. 172).

Salton und MacGill nennen folgende Faktoren als sinnvolle Aspekte zur Evaluation von IR-Systemen aus Sicht des Benutzers (vgl. Salton und MacGill 1987, S. 172):

- Recall,
- Precision,
- Abfall,
- Zeit,
- Aufwand,
- Präsentation und
- Abdeckung.

*Recall* und *Precision* beziehen sich auf die Menge der gefundenen und relevanten Objekte. *Recall* adressiert die Vollständigkeit der Ergebnisse und charakterisiert das Verhältnis zwischen den relevanten Objekten, die gefunden wurden und den relevanten Dokumenten aus der Gesamtheit der Dokumente. *Precision* steht für das Verhältnis zwischen der Anzahl der gefundenen relevanten Objekte zu der Anzahl aller gefundenen Dokumente, ist demnach ein Maß für die Genauigkeit/Treffsicherheit der Ergebnismenge. Als *Abfall* wird die Wahrscheinlichkeit bezeichnet, dass ein gefundenes Dokument nicht relevant ist.

Fuhr merkt an, dass vor allem bei Anwendungen, die statt einer Ergebnismenge eine Rangliste von Antworten ermitteln, häufig Varianten dieser Maße genutzt werden. Diese Varianten repräsentieren Maße, die an den jeweiligen Kontext angepasst wurden (vgl. Fuhr 2010, S. 13).

*„Beim Web-Retrieval kann man davon ausgehen, dass die meisten Benutzer (nach empirischen Untersuchungen ca. 90%) sich nur die erste Seite der Ergebnisliste anschauen, die in der Regel 10 Antworten enthält. Ein passendes Maß ist daher die Precision nach 10 Dokumenten, die meist als "Prec@10" bezeichnet wird. Ein extremer Standpunkt wäre die Precision des ersten Dokumentes (Prec@1). In diesem Kontext wird auch häufig die Click-Through-Rate betrachtet: in Ermangelung von Relevanzurteilen wird angenommen, dass jede angeklickte Antwort relevant ist.“ (Fuhr 2010, S. 13)*

*Zeit* bezeichnet die durchschnittliche Zeitdauer zwischen der Ausführung der Anfrage und der Präsentation des Ergebnisses. *Aufwand* kennzeichnet den Einsatz, den ein Benutzer bis zum Erhalt der Antwort aufbringen muss. Die Qualität der Ergebnisdarstellung wird als *Präsentation* bezeichnet, die *Abdeckung* ist ein Maß für das Vorhandensein relevanter Objekte im Ergebnis.

## 4.2 Informationsbedarfsanalyse

Das Ziel der Informationsbedarfsanalyse besteht darin, den Informationsbedarf und das Informationsbedürfnis zielgerichtet zu ermitteln. Sie bezieht sich auf die Art, die Qualität und die Menge der Information (vgl. Stelzer 2001). Während sich der Informationsbedarf bei gut strukturierten Arbeitsabläufen genau bestimmen lässt, ist eine Ermittlung bei schlechter Strukturierung der Abläufe aufwendiger und erfordert möglicherweise weitergehende Aktivitäten aus dem Bereich des Wissensmanagements (vgl. ebenfalls Stelzer 2001).

Es existieren unterschiedliche Verfahren zur Informationsbedarfsanalyse; nach Voß und Gutenschwager sind insbesondere die folgenden Methoden zu nennen (vgl. Voß und Gutenschwager 2001, 129ff.):



- das Literaturstudium,
- die Berichts-/Dokumentenanalyse,
- die Analyse bestehender Informationssysteme und
- die Expertenbefragung.

Ergänzend definieren Strauch und Winter folgende Analyseansätze im Rahmen der Ermittlung des Informationsbedarfs (vgl. Strauch und Winter 2002, 367f.):

- *Nachfrageorientierter Ansatz*: Berücksichtigung der Sicht des Benutzers und Ermittlung des subjektiven Informationsbedarfs (-bedürfnisses). Darunter können die Methoden des Literaturstudiums und der Expertenbefragung eingeordnet werden.
- *Angebotsorientierter Ansatz*: Identifizierung bestehender Informationssysteme und Ableitung des Informationsbedarfs. Darunter können alle vier beschriebenen Methoden eingeordnet werden.
- *Geschäftsprozessorientierter Ansatz*: Analyse bestehender Geschäftsprozesse im Hinblick auf die Informationsnutzung. Darunter können ebenfalls alle vier beschriebenen Methoden eingeordnet werden

In Anlehnung an Krcmar (vgl. Krcmar 2010) und Hauschildt (vgl. Hauschildt 1990) klassifiziert Wolf die verschiedenen Verfahren zur Ermittlung des Informationsbedarfs in die Kategorien *subjektive Verfahren*, *objektive/analytische Verfahren* und *theoriegestützte/ empirisch fundierte Verfahren* (Wolf 2007, S. 46). Tabelle 3 fasst die Verfahren zusammen.

Verfahren zur Ermittlung des Informationsbedarfs		
Subjektive Verfahren	Objektive, analytische Verfahren	Theoriegestützte, empirisch fundierte Verfahren
<ul style="list-style-type: none"> <li>• Offene Befragung</li> <li>• Teilnehmende Beobachtung</li> <li>• Wunschkataloge</li> <li>• Befragung der Mitarbeiter im Tätigkeitsumfeld</li> </ul>	<ul style="list-style-type: none"> <li>• Strategieanalyse</li> <li>• Prozessanalyse</li> <li>• Input-Prozess-Output-Analyse</li> <li>• Entscheidungsanalyse</li> </ul>	<ul style="list-style-type: none"> <li>• Strukturierte Befragung</li> <li>• Methode der kritischen Erfolgsfaktoren</li> <li>• Balanced Scorecard</li> <li>• Weiterentwicklung aus dem Kontext (Evolution)</li> <li>• Entwicklung aus dem Bestehenden (Ist-Situation bzw. Tätigkeitsanalysen)</li> </ul>

**Tabelle 3: Verfahren zur Ermittlung des Informationsbedarfs**

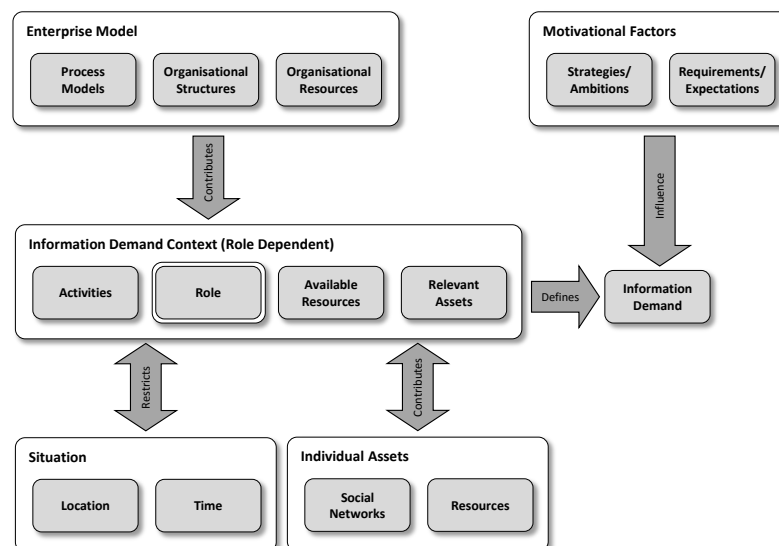
Quelle: (Wolf 2007, S. 46; Hauschildt 1990, S. 527 und Krcmar 2010, S. 65)

Die unterschiedlichen Methoden zur Informationsbedarfsanalyse adressieren jeweils unterschiedliche Domänen, insbesondere verschiedene betriebswirtschaftliche Domänen. Jedoch lassen sich diese Methoden auch auf den in dieser Arbeit fokussierten Informationsbedarf eines einzelnen Benutzers im Rahmen seiner Arbeit mit Anwendungssystemen übertragen; darauf wird im weiteren Verlauf der Arbeit näher eingegangen.

Die Berücksichtigung von Kontext im Rahmen des Informationsbedarfs definiert Lundqvist wie folgt:

*“The Information Demand of an entity, may it be an individual or an organisation, is defined by that entity’s Context. The Context comprises the Role the entity has within a larger organisational structure, the Activities performed by the Role and any Resources available to it for doing so. Whilst parts of a Context can be derived from Enterprise Models additional aspects affecting the Context such as Individual Assets and Situations contributes to and restricts the Context in terms of susceptibility and availability to formal or informal information. The Information Demand of the user can be further influenced by different Motivational Factors adding to the demand dictated by the Context.” (Lundqvist 2007, S. 67)*

Abbildung 42 visualisiert die Zusammenhänge der unterschiedlichen Bestandteile der Definition in einer integrierten Sicht des Informationsbedarfs. Es wird deutlich, dass der Informationsbedarf durch den Kontext des Informationsbedarfsträgers definiert wird, wobei die Rolle des Bedarfsträgers, seine Situation und seine persönlichen Faktoren bestimmend sind. Zusätzlich beeinflussen Aspekte der Motivation den Informationsbedarf.



**Abbildung 42: Integrierte Sicht des Informationsbedarfs**

Quelle: (Lundqvist 2007, S. 68)

### 4.3 Benutzerverhalten innerhalb des IR

Wie zuvor erläutert (vgl. Kapitel 1) ist das Ziel des intendierten Frameworks die Bereitstellung von Informationen für Endbenutzer, die in softwaregetriebenen Umgebungen agieren. In diesem Abschnitt wird erläutert, wie solche Endbenutzer nach Informationen suchen. Dazu wird exemplarisch auf das Benutzerverhalten im Rahmen von Suchprozessen im Web eingegangen, denn das Web mit seiner riesigen Informationsbasis wird von einer Vielzahl Endbenutzer intensiv zur Informationssuche verwendet.

Die wissenschaftliche Disziplin der *Benutzerforschung* befasst sich mit der Analyse des Benutzungs- und Informationsverhaltens von Individuen:

*„Die Benutzerforschung versucht, durch die Analyse des Benutzungs- und Informationsverhaltens Aussagen über Struktur und Eigenschaften eines in einem Systemzusammenhang stehenden Handlungssubjektes abzuleiten, um dadurch das gesamte Aktionsgefüge – d. h. Institution und Benutzer – effizienter gestalten zu können.“ (Koreimann 1976, S. 72)*

Im Rahmen der Benutzerforschung wird der Benutzer als Informationsbedarfsträger stets im Zusammenhang mit seiner Umwelt betrachtet. Nach Wersig werden vier unterschiedliche Typen von Benutzern unterschieden: *Potenzielle Benutzer*, *vermutete Benutzer*, *tatsächliche Benutzer* und *Nutzer* (Wersig 1973, S. 12). Potenzielle Benutzer bilden den Typ Benutzer, die möglicherweise Interesse an einer bestimmten Information besitzen. Benutzer, die Zugang zu dieser Information haben, werden als vermutete Benutzer bezeichnet, diejenigen, die die Information nutzen, als tatsächliche Benutzer. Die tatsächlichen Benutzer, die Nutzen aus dieser Information ziehen bzw. von ihr profitieren, werden als Nutzer (der Information) bezeichnet.

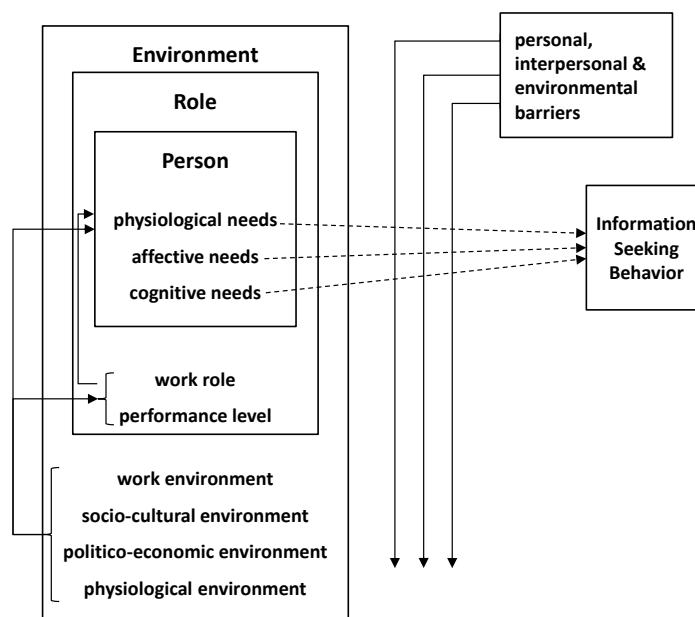
Des Weiteren werden die Benutzer typologisiert, um „*Gemeinsamkeiten des Informationsbedarfs einer potenziellen Menge von Benutzern zu erkennen, um davon ausgehend ein generelles ‚Informationsangebot‘ definieren zu können*“ (Koreimann 1976, S. 78).

Wersig unterscheidet folgende Analysen für die Benutzerforschung (Wersig 1973, S. 13):

- *Informationsflussanalysen*: sie ermitteln die Struktur der Kommunikation bzw. des Informationsaustauschs.
- *Benutzeranalysen*: sie untersuchen Benutzerrollen und -gruppen.
- *Benutzerforschung im engeren Sinn*: sie analysiert das Problemlösungsverhalten.
- *Nutzungsanalysen*: sie untersuchen die subjektive Informationsbewertung.

Zum Informationsverhalten der Benutzer unterscheidet Wilson drei Perspektiven (vgl. Wilson 1981): Das Informationsverhalten (engl. *information behaviour*) beschreibt

generell die Gesamtheit des menschlichen Verhaltens in Beziehung zu Informationsquellen und -kanälen. Informationsrechercheverhalten (engl. *information seeking behaviour*) bezeichnet eine zielgerichtete Recherche nach Informationen als Konsequenz des Bedürfnisses, einen Bedarf zu decken. Das Informationssuchverhalten (engl. *information search behaviour*) bezeichnet die explizite Interaktion des Benutzers mit einem Informationssystem. Insbesondere das Rechercheverhalten ist Betrachtungsgegenstand in Wilson's Modell des Informationsverhaltens; es wird in Abbildung 43 dargestellt.



**Abbildung 43: Modell des Informationsverhaltens**

Quelle: (Wilson 1981, S. 8)

Das Modell von Wilson lässt sich auf den Gegenstand der vorliegenden Arbeit übertragen: Auch im Kontext von PERCOP agiert eine Person innerhalb einer Umgebung und besitzt dabei eine bestimmte Rolle. Es entstehen unterschiedliche Informationsbedarfe, die zu einem Informationsrechercheverhalten führen. Der Prozess der Informationsrecherche ist mit verschiedenen Barrieren behaftet, die es zu überwinden gilt.

### Suchstrategien

Die Tatsache, dass die Informationsbasis des Web extrem groß und nicht einheitlich erfasst bzw. indiziert ist, führt zu entsprechend aufwendigen und komplizierten Suchaktivitäten. Suchmaschinen dienen zur Suche nach relevanten Informationen; sie liefern allerdings nur dann nützliche Informationen, wenn eine sinnvolle Suchstrategie verfolgt wird. Es verwundert deshalb nicht, dass bei einer Befragung 83% der Befragten zu der Frage „Wie wichtig werden folgende Schlüsselqualifikationen zukünftig für die

*tägliche Internetnutzung sein?“, die Antwort „Wissen, wie man Informationen sucht“ auswählten und 82% der Befragten „die Fähigkeit, die Relevanz von Informationen zu beurteilen“<sup>37</sup>.*

Eine Vielzahl von Studien hat sich mit dem Suchverhalten von Personen insbesondere im Web befasst. Sie haben u. a. ergeben, dass Suchmaschinen unterschiedlich und häufig nicht optimal genutzt werden (vgl. u. a. Fries 2007, Hölscher und Strube 2000 sowie Höchstötter 2007). Das ergibt sich daraus, dass die Möglichkeiten, die die Suchmaschinen bieten, sehr häufig nicht ausgenutzt werden; beispielsweise bleiben Verknüpfungsoperatoren (UND, ODER, NICHT), aber auch Sprachoptionen und Synonyme ungenutzt.

Tabelle 4 zeigt ein entsprechendes Ergebnis einer Untersuchung zur Nutzung von Formatierungsfunktionen einer Suchmaschine einerseits durch „Experten“ und andererseits durch „Anfänger“.

	Domain knowledge LOW		Domain knowledge HIGH	
	Web novice	Web expert	Web novice	Web expert
AND	4.2	25.9	-	11.1
+ (plus)	33.3	44.4	11.1	22.2
“ ”	20.8	22.2	18.5	37.0
any kind of formatting	58.3	92.6	37.0	81.5
Errors	29.2	-	11.1	3.7

**Tabelle 4: Abfrage-Formatierung in simulierten Suchen (Prozent)**

Quelle: (Hölscher und Strube 2000)

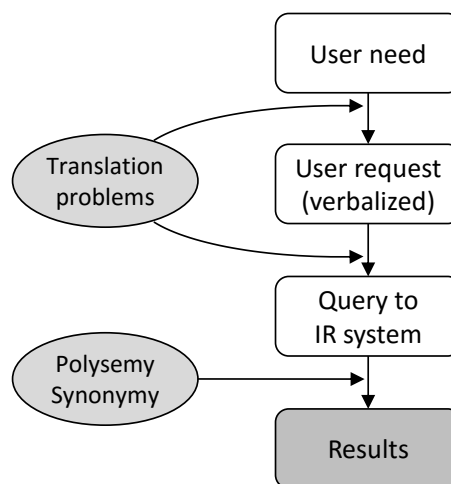
Die Ergebnisse der Studie (vgl. Hölscher und Strube 2000) zeigen, dass nur ein sehr geringer Anteil der unerfahrenen Nutzer andere Suchoptionen und andere Suchoperatoren als + und „“ (Anführungszeichen) verwendet. Zudem ist die Länge der Suchanfragen im Durchschnitt sehr gering: Anfänger nutzten durchschnittlich nur 2,32 Wörter, Experten sind mit durchschnittlich 2,61 Wörtern nicht wesentlich „besser“. Als ein signifikanter Faktor bezüglich der Länge der Suchanfrage wurde durch die Studie das vorhandene Domänenwissen ermittelt: Teilnehmer mit geringen Domänenkenntnissen setzten bei ihrer Suche im Durchschnitt ca. 1 Wort mehr pro Abfrage ein. In den letzten

<sup>37</sup> Quelle: <http://de.statista.com/statistik/daten/studie/166379/umfrage/schlüsselqualifikationen-fuer-die-internetnutzung/>

Jahren ist die durchschnittliche Anzahl der Wörter einer Suchanfrage zwar gestiegen, jedoch nutzt ein großer Teil der Benutzer (insbes. Laien) weiterhin lediglich ein Wort für die Suche<sup>38</sup>. Auch die Möglichkeiten, unterschiedliche Sprachen zu verwenden wird häufig nicht genutzt.

Des Weiteren wurde festgestellt, dass der Großteil der Anwender die Suchabfrage nicht ändert („78% of queries are not modified“), und häufig nur die erste Ergebnisübersicht betrachten („85% look over one result screen only“) (vgl. Boumans und Bruck 2008, S. 124). Dieses sog. „Suchdilemma“ (vgl. Abbildung 44) betrifft alle Aktivitäten zwischen der Feststellung des Bedarfs eines Benutzers und der Präsentation der Suchergebnisse; es führt zu einer großen Anzahl nicht relevanter respektive nicht pertinenter Ergebnisse.

Das „Suchdilemma“ bei einem WEB-IR-Prozess entsteht insbesondere durch zwei Problemklassen. Zum einen haben Benutzer Schwierigkeiten bei der verbalen Formulierung ihres Informationsbedarfs (*Translation problems*) und bei der Erstellung der Anfrage an das IR-System. Zum anderen führen Polysemie<sup>39</sup> und Synonymie<sup>40</sup> zu Problemen (vgl. Henzinger 2008).



**Abbildung 44: Suchdilemma bei einem Web-IR-Prozess**

Quelle: (Henzinger 2008)

<sup>38</sup> Quelle: <http://www.newmediatrendwatch.com/markets-by-country/17-usa/123-demographics?start=4>:

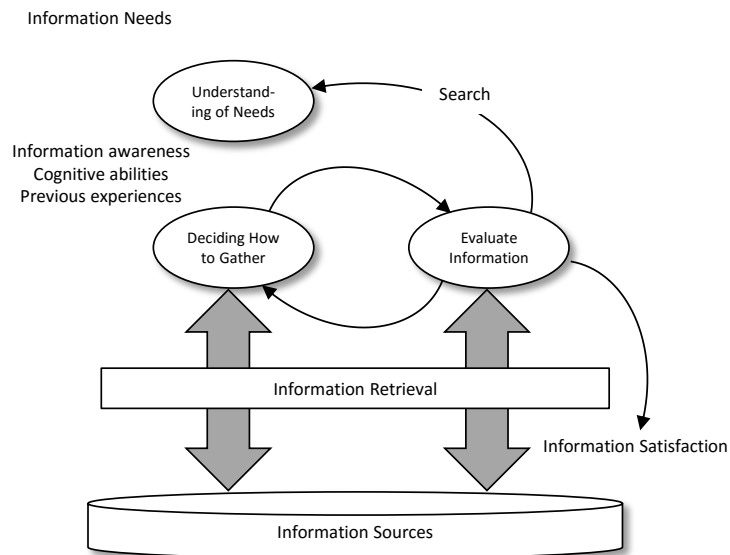
1 Wort: 24.34%, 2 Wörter: 23.41%, 3 Wörter: 20.32%, 4 Wörter: 13.79%, 5 Wörter: 8.02%.

6 Wörter: 4.37%, 7 Wörter: 2.37%, 8 oder mehr Wörter: 3.38%

<sup>39</sup> Polysemie bezeichnet sprachwissenschaftlich die Mehrdeutigkeit eines Wortes (Zeichen). Das Wort „Jaguar“ kann beispielsweise ein Auto, jedoch auch ein Tier bezeichnen. Eine „Bank“ kann ein Möbelstück, jedoch auch ein Geldinstitut sein.

<sup>40</sup> Synonymie bezeichnet sprachwissenschaftlich die gleiche Bedeutung unterschiedlicher Wörter (Zeichen), beispielsweise bei den Wörtern „Schädel“ und „Kopf“.

Abbildung 45 zeigt das konzeptuelle Modell der Informationsrecherche. Das Modell verdeutlicht insbesondere, dass der Rechercheprozess durch die kognitiven Fähigkeiten, dem Informationsbewusstsein und vorherigen Erfahrungen des Informationsbedarfsträgers beeinflusst wird. Des Weiteren macht das Modell deutlich, dass dem Informationsbedarfsträger sein Bedarf nach Informationen bewusst sein muss.



**Abbildung 45: Konzeptuelles Modell der Informationsrecherche**

Quelle: In Anlehnung an (Jentzsch und Prekop 2001, S. 100)

## 4.4 Information Retrieval innerhalb von PERCOP

Um eine hohe Qualität bei der Informationsbereitstellung zu erreichen, sollen innerhalb von PERCOP die unterschiedlichen Ausprägungen von Relevanz berücksichtigt werden. Insbesondere soll Pertinenz durch das Feedback der einzelnen Benutzer adäquat realisiert werden.

Die Qualität des Feedbacks kann gesteigert werden, indem zu einer Information bzw. zu einem verlinkten Dokument nicht nur eine einfache „Gut-oder-schlecht-“ bzw. „Hilft-hilft nicht-“ Rückmeldung erfragt wird, sondern eine detailliertere Rückmeldung ermöglicht wird. Als einfaches Beispiel seien die nachfolgenden Fragen genannt, die jeweils mit ja oder nein zu beantworten sind. Die Fragen beziehen sich auf die initiale Fragestellung „Warum ist das Dokument nicht nützlich?“:

- „Ich kenne die Information bereits!“,
- „Die Information liefert keine Lösung für das Problem!“ oder
- „Ich verstehe die Information nicht!“.

Ein so spezifiziertes Feedback kann innerhalb des Frameworks dazu genutzt werden, dass die einzelnen Recommender Informationsdokumente ermitteln, die für den jeweiligen Benutzer eine höhere Relevanz besitzen: Das Feedback „*Ich kenne die Information bereits*“ ist wichtig, da diese Aussage die objektive Relevanz der Information nicht verändert. Die Aussage „*Ich verstehe die Information nicht*“ kann nützliche Hinweise zum aktuellen Profil des Benutzers liefern.

Objektive Relevanz soll durch die Aggregation aller Feedbacks und durch Bewertungen der Crowd erfolgen. Des Weiteren soll das System auf Basis aller vorhandenen Informationen automatisch eine Relevanzbeurteilung vornehmen (Systemrelevanz).

Die unterschiedlichen IR-Modelle werden innerhalb der PERCOP-Recommender implementiert. Die Empfehlungsmodule können die unterschiedlichen Kontextinformationen nutzen, um relevante Informationen zu ermitteln. Beispielsweise können Web-Retrieval-Modelle implementiert werden, die die Sprache des Benutzers, Synonyme und Korrekturen berücksichtigen

Bezogen auf die Strategie fokussiert PERCOP insbesondere den ungedeckten Informationsbedarf innerhalb der Push-Strategie. Die Strategie ermöglicht es, einem Benutzer ungefragt Informationen bereitzustellen. Bei Nutzung der Pull-Strategie müsste sich der Benutzer zum einen eines Informationsdefizits bewusst sein, zum anderen wäre seine aktive Beteiligung erforderlich. Das Ziel von PERCOP besteht jedoch darin, dem Benutzer Informationen automatisiert bereitzustellen und ihm die Formulierung der jeweiligen Suchanfrage „abzunehmen“.

Die Informationsbedarfsanalyse adressiert Projekte bzw. auch Vorgänge, in denen es darum geht, Personen alle für eine bestimmte Aufgabe benötigten Informationen in der richtigen Form zur richtigen Zeit zur Verfügung zu stellen. Für PERCOP besteht die Möglichkeit, Ansätze und Methoden der Informationsbedarfsanalyse zu adaptieren. Denn das Framework adressiert insbesondere das subjektive Informationsbedürfnis des einzelnen Benutzers. Aus den Aufzeichnungen der Benutzeraktivitäten lassen sich in Kombination mit den ermittelten und bewerteten Informationen nützliche Informationen über den objektiven Informationsbedarf ermitteln. Bei diesem Vorgehen können Methoden des *geschäftsprozessorientierten Ansatzes* Verwendung finden.

Des Weiteren wird der Bedarf durch die Erstellung von Regeln und der Zuordnung von Informationen zu Funktionalitäten durch Teilnehmer der Crowd bestimmt. Dieses Verfahren entspricht grundsätzlich dem Ansatz der *Expertenbefragung*.



Wie auch bei den Verfahren der Benutzerforschung soll die Typologisierung der Benutzer durch die Verwendung von Profilinformationen erfolgen. Eine Typologisierung ermöglicht die Bildung von Cluster ähnlicher Benutzer, wodurch wiederum auf einen vergleichbaren Informationsbedarf geschlossen werden kann.

Des Weiteren soll das PERCOP-Framework die Integration unterschiedlicher Analysen des Nutzungsverhaltens ermöglichen, beispielsweise durch die Automatisierung der Erfassung von Kontextinformationen in Kombination mit den Bewertungen der Benutzer zu den erhaltenen Informationen.

Vergleichbar mit dem Rahmenmodell für visuell direkt-manipuliertes Information Retrieval von Landwiche (vgl. Abbildung 38) sollen auch innerhalb von PERCOP die einzelnen Phasen mit dem Ziel durchlaufen werden, einen möglichst hohen Automatisierungsgrad zu realisieren:

*Fokus/Interesse* kann dabei in unterschiedlicher Form realisiert werden: Durch die Push-Strategie kann dem Benutzer aktiv eine Liste potenziell relevanter Informationen präsentiert werden; der Fokus wird bei dieser Strategie automatisch gesetzt. Des Weiteren kann das Interesse an einer Information einem Benutzer auch „aufgezwungen“ werden, beispielsweise, wenn durch eine konkrete Regel innerhalb des PERCOP-Frameworks eine Information exakt einer Benutzeraktion zugewiesen wird.

Die *Informationsvisualisierung* soll einheitlich durch eine zentrale Client-Anwendung bei dem Benutzer erfolgen. Die *Wahrnehmung* der Informationen durch den Benutzer soll dahingehend beeinflusst werden, dass zum einen eine unauffällige Bereitstellung der Informationen realisiert wird. Dann wird der Benutzer bei seiner Arbeit nicht abgelenkt, kann selbst entscheiden, ob er die angebotenen Informationen nutzen möchte.

Zum anderen kann für eine besonders wichtige/relevante Information die Aufmerksamkeit des Benutzers durch eine auffällige Präsentationsart geweckt werden. Die unterschiedlichen Formen der Präsentation ermöglichen auch eine Integration der Informationsversorgung in Geschäftsprozesse. Eine detaillierte Betrachtung einschließlich der Darstellung weiterer Beispiele erfolgt im weiteren Verlauf der Arbeit bei der Entwicklung des Prototyps in Kapitel 8 und bei der Vorstellung der Fallbeispiele in Kapitel 9.

Wie bereits in Abschnitt 1.2 erläutert, besteht ein Ziel des intendierten Frameworks darin, den manuellen Aufwand für die Benutzer so niedrig wie möglich zu gestalten. Daher ist eine sehr einfache und schnelle manuelle Bewertung der Informationen innerhalb von PERCOP anzustreben, worauf in Abschnitt 6.7 detailliert eingegangen wird. Aber auch auf

Basis von Verweildauern der Benutzer bei der Betrachtung der Informationsdokumente oder durch die konkrete Wahl eines Dokuments kann auf unterschiedliche Kriterien der Evaluation geschlossen werden. Die Berücksichtigung der Dauer bei der Ermittlung und Darstellung von Informationen ist wichtig, da das Framework auf situationsaktuelle Darstellung der Informationen ausgelegt ist.

Die Entscheidung über die Art der Suche soll durch das PERCOP-Framework übernommen werden, indem Umgebungs- und Profilinformatoren berücksichtigt werden. Die *Evaluierung* erfolgt weiterhin durch den Benutzer. Durch das Framework können jedoch auch andere Benutzer von der Evaluierung eines Benutzers profitieren, indem die individuellen Bewertungen der Informationen zusammen mit der verwendeten Recommender-Konfiguration und dem Benutzerprofil für die Informationsermittlung aller Benutzer verwendet wird.

PERCOP lässt sich auf das konzeptuelle Modell der Informationsrecherche von Jentzsch und Prekop übertragen (vgl. Abbildung 45): Zunächst wird der Bedarf automatisch ermittelt; die Ermittlung erfolgt zum einen auf der Basis der Beobachtung der Benutzeraktivitäten, zum anderen durch die Zuordnung der relevanten Informationen entweder direkt durch die Mitglieder der Crowd oder automatisch durch Ausführung von Regeln im Rahmen der Recommender/-Konfigurationen.

Die Verwendung von Information Retrieval innerhalb von PERCOP soll an folgendem Beispiel verdeutlicht werden: Wie bereits zuvor erläutert, ermöglichen Suchmaschinen die Verwendung von erweiterten Optionen für die Erstellung einer Suchanfrage. Dazu gehören beispielsweise:

- *Boolsche Operatoren*: Ermöglichen das Ein-/Ausschließen einzelner Bestandteile (AND, OR, NOT, +, -).
- *Phrasen*: Suche nach exakten Wortkombinationen, häufig durch Anführungsstriche markiert.
- *Plural, Großbuchstaben, alternative Schreibweisen*: Erweiterung oder Eingrenzung bestimmter Begriffe.
- *Synonyme*: Ergänzung von Wörtern gleicher Bedeutung.
- *Rechtschreibung*: Überprüfung oder automatische Korrektur.
- *Sprache*: Eingrenzung auf konkrete Sprachen bzw. automatische Suche in anderen Sprachen.
- *Suchort*: Bestimmung des Bereichs, innerhalb dessen die Suche stattfindet (z. B. URLs, Domains, Hosts).

Die erweiterten Optionen werden (wie bereits erläutert) nur unzureichend durch Benutzer verwendet). Ziel des PERCOP-Frameworks in Hinblick auf die Nutzung von Suchmaschinen ist es daher nicht nur, die Erstellung der Anfragen zu automatisieren, sondern auch durch die Verwendung der erweiterten Möglichkeiten zur Anfrageformulieren die Qualität der Suche zu verbessern.

Des Weiteren können Benutzer eines PERCOP-Frameworks von den Kenntnissen anderer Anwender profitieren. Mitglieder der Crowd, die als Experte in der Lage sind, „gute“ Suchabfragen zu formulieren, können diese im Rahmen des Frameworks bereitstellen, so dass diese Anfragen auch im Rahmen der automatisierten Suche für andere Anwender genutzt werden können.

## 5 Nutzung des Potenzials der Masse

Eine wesentliche Eigenschaft des in dieser Arbeit beschriebenen PERCOP-Frameworks ist die Einbeziehung der *Masse* (*Crowd*) als eine unbestimmte Anzahl unterschiedlicher Personen. In *The Wisdom of the Crowds* beschäftigt sich Surowiecki ausführlich mit der sog. *Weisheit der Vielen*. Er definiert folgende Kriterien, welche „weise“ Massen (*Crowds*) von irrationalen unterscheiden (vgl. Surowiecki 2005):

- *Diversity of opinion (Meinungsvielfalt)*: Jede Person besitzt eigene Informationen, auch wenn diese eventuell nur persönliche Interpretation bekannter Fakten sind.
- *Independence (Unabhängigkeit)*: Die Meinung des Einzelnen ist nicht bestimmt durch die Meinung anderer.
- *Decentralization (Dezentralisierung)*: Die Menschen sind in der Lage zu spezialisieren und eigenes Wissen anzuwenden.
- *Aggregation (Aggregation)*: Es existieren Mechanismen, die aus den Meinungen der Einzelnen Gruppenentscheidungen bilden.

Insbesondere seit der Entstehung des *Web 2.0* entstanden unterschiedliche Disziplinen, welche sich mit der Nutzung von verteilter Intelligenz beschäftigen. In diesem Kapitel werden zunächst die für die vorliegende Arbeit relevante Disziplinen *Kollektive Intelligenz*, *Crowdsourcing*, *Interaktive Wertschöpfung* und *Open Source* beschrieben. Anschließend werden die Disziplinen in Hinblick auf das Ziel der Arbeit analysiert und bewertet.

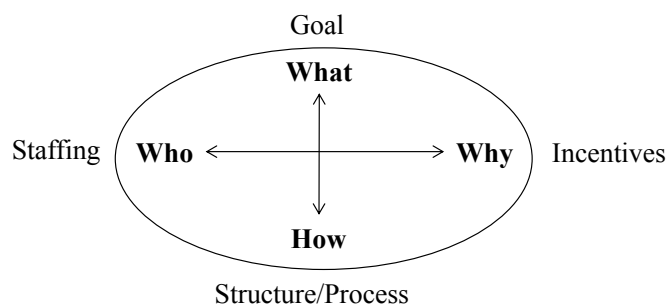
### 5.1 Kollektive Intelligenz

*Kollektive Intelligenz* bildet den Oberbegriff über die unterschiedlichen Ausprägungen der Nutzung von Massenintelligenz und wird in unterschiedlichen Disziplinen seit geraumer Zeit untersucht (vgl. Leimeister 2010b, S. 239). Lévy definiert kollektive Intelligenz wie folgt:

*“[...] a form of universally distributed intelligence, constantly enhanced, coordinated in real time, and resulting in the effective mobilization of skills. [...] The basis and goal of collaborative intelligence is the mutual recognition and enrichment of individuals rather than the cult of fetishized or hypostatized communities.” (Lévy 1997, S. 13)*

In dieser Definition werden insbesondere die Eigenschaften der Verteilung, der ständigen Verbesserung und der Echtzeitverarbeitung hervorgehoben. Insbesondere das *MIT Center of Collective Intelligence* befasst sich mit der Thematik und definiert den Begriff kurz mit: *“groups of individuals doing things collectively that seem intelligent”* (Malone et al. 2009, S. 1).

Die Entwicklung des Internets hat zu neuen Möglichkeiten der Nutzung kollektiver Intelligenz geführt. *“Collective Intelligence has received a new meaning in recent years, especially through the emergence of new (mostly Web 2.0) applications and user generated content”* (Leimeister 2010a). Peerce und Shneiderman bezeichnen diesen Trend als *Technology-Mediated Social Participation* (vgl. Peerce und Shneiderman 2009). Gemeint ist damit *„die Fähigkeit der Massen, durch Partizipation im Internet gemeinschaftlich Ziele zu erreichen, die einzelne Individuen oder auch Organisationen selbst nicht leisten können.“* (Leimeister 2010b, S. 240)



**Abbildung 46: Elements of collective intelligence building blocks**

Quelle: (Malone et al. 2009, S. 3)

Malone et al. definieren einen Satz von Grundbausteinen, den sog. *Building Blocks*, die für den Großteil von Collective-Intelligence-Systemen von Relevanz sind. Als Analogie zur Biologie bezeichnen Malone et al. die Grundbausteine als „Gene“ (vgl. Malone et al. 2009).

Vier Hauptfragen gilt es zu beantworten, um diese Gene zu klassifizieren (siehe Abbildung 46):

- Wer führt die Aufgabe durch?
- Welche Motivation steckt dahinter?
- Welche Tätigkeit wird durchgeführt?
- Wie wird diese Tätigkeit durchgeführt?

Im Hinblick auf das in dieser Arbeit vorgestellte Framework können die Kernfragen wie folgt beantwortet werden:

*Wer führt die Aufgaben durch?* - Malone et al. unterscheiden das sog. *Hierarchy-* und das *Crowd-Gen*. Bei dem Hierarchy-Gen entscheidet eine höhere Instanz über die Verteilung der Aufgaben, während bei dem Crowd-Gen die Individuen selbständig und gleichberechtigt entscheiden. Auch innerhalb von PERCOP soll die Crowd eigenständig

handeln; sie wird repräsentiert durch eine unabhängige Masse von Menschen. Die Beteiligung der einzelnen Individuen kann in unterschiedlicher Art und Weise geschehen, abhängig von der jeweiligen Rolle, die das Individuum im konkreten Kontext und innerhalb der jeweiligen Domäne besitzt.

*Welche Motivation steckt dahinter?* - Nach Malone et al. sind die Hauptgründe für die Beteiligung an Collective-Intelligence-Systemen *Money* (Geld), *Glory* (Ruhm) oder *Love* (Liebe) (Malone et al. 2009, S. 5). Neben dem finanziellen Aspekt bei der Beteiligung spielen heutzutage immer häufiger die beiden anderen Gründe eine große Rolle. Sowohl Spaß an der Beteiligung (Love) als auch Respekt und Anerkennung (Glory) anderer für die geleistete Arbeit sind wesentliche Treiber bei Collective-Intelligence-Systemen (vgl. Malone et al. 2009, S. 5).

*Welche Tätigkeit wird durchgeführt?* - Die Art der Tätigkeiten wird differenziert in *Erstellen* und *Entscheiden* (*Create/Decide*). Im ersten Fall erzeugen die Akteure des Systems etwas „Neues“, im zweiten werden Alternativen evaluiert und Entscheidungen getroffen.

Hauptziel von PERCOP ist die Verbesserung der Informationsbeschaffung im Hinblick auf Menge, Qualität und Relevanz der Informationsartefakte in einem konkreten Kontext. Dazu soll das Framework eine Vielzahl von Beteiligungsschnittstellen anbieten (siehe Kapitel 3).

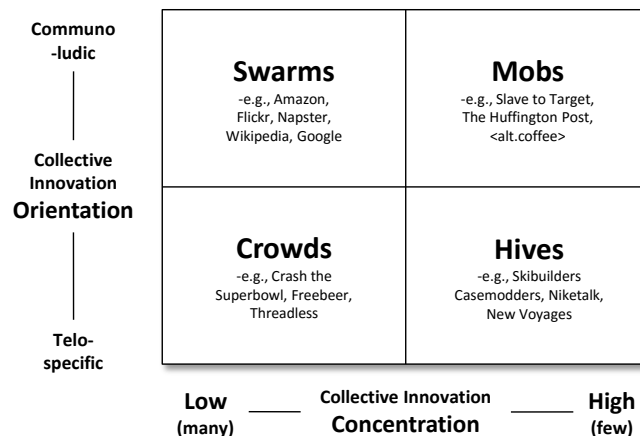
*Wie wird diese Tätigkeit durchgeführt?* - Bei der Frage nach dem *Wie* wird unterschieden zwischen unabhängigen und abhängigen Tätigkeiten sowie zwischen Erzeuger- und Entscheider-Tätigkeiten. Bei erzeugenden Aktivitäten entstehen neue Ergebnisse in Form von *Collection* (bei unabhängigen Tätigkeiten) oder *Collaboration* (bei abhängigen Tätigkeiten). Entscheider-Tätigkeiten sind entweder individuell (bei unabhängigen Tätigkeiten) oder gruppengetrieben (bei abhängigen Tätigkeiten).

Über die PERCOP-Plattform sollen die einzelnen Individuen der Crowd in der Lage sein, sich an dem gemeinsamen Ziel zu beteiligen. Die Plattform soll die Einzelaktivitäten zusammenführen. Dadurch können beispielsweise Zuordnungen erstellt, Regeln definiert oder Sensoren oder Empfehlungsmodule integriert werden.

Tapscott und Williams haben den Begriff *Wikinomics* geprägt für die Möglichkeit der Massenkollaboration durch die Entstehung neuer Technologien (Web 2.0) und Entwicklungen wie der Net Generation und der Globalisierung (vgl. Tapscott und Williams 2007). Sie definieren folgende vier Prinzipien:

- *Being Open*: Offenheit in Hinblick auf neue Ideen und Konzepte.
- *Peering*: Horizontale Organisationsstrukturen mit selbstorganisierenden Teams.
- *Sharing*: Transparenz bei der Bereitstellung von Informationen.
- *Acting globally*: Global agieren, Nutzerkreis vergrößern.

Die unterschiedlichen Einsatzgebiete und Möglichkeiten zur Beteiligung im Rahmen kollektiver Systeme haben dazu geführt, dass die Masse verschiedene Eigenschaften aufweisen kann. Kozinets et al. identifizieren im Rahmen einer Typologie von Online Creative Consumer Communities vier unterschiedliche Arten der Masse bzw. von Online Communities: *Swarms*, *Mobs*, *Crowds* und *Hives* (vgl. Kozinets et al. 2008, S. 345). Abbildung 47 stellt die vier Arten mit Bezug einerseits zum Grad der *Collective Innovation Orientation* und andererseits der *Collective Innovation Concentration* grafisch dar.



**Abbildung 47: Typologie von Online Creative Consumer Communities**

Quelle: (Kozinets et al. 2008, S. 345)

*Swarms* sind eine sehr große Masse von Individuen, bei der das einzelne Individuum nur einen minimalen Beitrag leistet. Das Ergebnis eines Swarms insgesamt ist jedoch von hohem Wert. Genutzt wird diese Form der Massenintelligenz insbesondere in Bewertungs-, Empfehlungssystemen und Tagging-Communities<sup>41</sup>.

*Hives* bezeichnen Online-Communities mit Individuen, die jeweils über hohe Fähigkeiten verfügen und in ihrer Gesamtheit auch innovative Ergebnisse erzielen können. Die einzelnen Mitglieder "*simply enjoy socializing over an ambitious project, testing and building their abilities, meeting a challenge*" (Kozinets et al. 2008, S. 347). Aus diesem Grund basieren viele Open-Source-Software-Projekte auf Hives-Communities.

<sup>41</sup> Konkrete Beispiele finden sich in (Kozinets et al. 2008, 350ff.)

Mit *Mobs* werden kleinere Gruppen bezeichnet, in denen die Individuen gleiche Interessen besitzen und sich über diese austauschen; die gemeinsamen Interessen bzw. Vorlieben stehen im Vordergrund. Zu den konkreten Themen steuern die Mitglieder Expertenwissen bei, wodurch innovative Ergebnisse produziert werden können.

*Crowds* charakterisieren Kozinets et al. als „*large, organized groups who gather or are gathered together specifically to plan, manage, and/or complete particular tractable and well defined projects*“ (Kozinets et al. 2008, S. 345). Sie grenzen die Crowd von den anderen Arten der Masse insbesondere dadurch ab, dass nur eine geringe Konzentration auf kollektive Innovationen besteht, und dass sich Crowds vorzugsweise an einzelnen bzw. speziellen Projekten beteiligen. Die Individuen einer Crowd beteiligen sich primär zu ihrem eigenen Vergnügen (siehe Abschnitt 5.6.3).

Bei der Konzeption des Frameworks kann die Unterscheidung der Arten der Masse zur Adaption vorhandener Erkenntnisse genutzt werden. Beispielsweise können für die einzelnen Bereiche der Beteiligung die relevanten Arten und die adressierte Domäne analysiert und in das zu entwickelnde Framework integriert werden.

## 5.2 Crowdsourcing

Im Kontext kollektiver Intelligenz hat das Konzept des sog. *Crowdsourcing* (CS) in den letzten Jahren stark an Beliebtheit gewonnen. Erstmals im Jahr 2006 von Jeff Howe beschrieben, bezeichnet Crowdsourcing „[...] *the act of a company or institution taking a function once performed by employees and outsourcing it to an undefined (and generally large) network of people in the form of an open call*“ (Howe 2006). Tätigkeiten, die bisher von speziell vorgesehene Personen (in der Regel Mitarbeiter einer Organisation) durchgeführt wurden, werden beim Crowdsourcing über einen offenen Aufruf an eine undefinierte große Gruppe Individuen übertragen. Papsdorf entwirft eine ähnliche Definition:

*"Crowdsourcing ist die Strategie des Auslagerns einer üblicherweise von Erwerbstätigen entgeltlich erbrachten Leistung durch eine Organisation oder Privatperson mittels eines offenen Aufrufes an eine Masse von unbekannten Akteuren, bei dem der Crowdsourcer und/oder die Crowdsourcees frei verwertbare und direkte wirtschaftliche Vorteile erlangen."* (Papsdorf 2009, S. 69)

Martin et al. sehen in der Definition von Howe ein starkes Ungleichgewicht zwischen dem theoretischen Verständnis und der unternehmerischen Realität, kritisieren insbesondere, dass die notwendigen geeigneten IuK-Technologien (z. B. Web 2.0) nicht angesprochen werden. Sie schlagen daher eine erweiterte Definition des Crowdsourcing-Begriffs vor:



*„Crowdsourcing ist eine interaktive Form der Leistungserbringung, die kollaborativ oder wettbewerbsorientiert organisiert ist und eine große Anzahl extrinsisch oder intrinsisch motivierter Akteure unterschiedlichen Wissensstands unter Verwendung moderner IuK-Systeme auf Basis von Web 2.0 einbezieht. Leistungsobjekt Dienstleistungen unterschiedlichen Innovationsgrades, welche durch das Netzwerk der Partizipierenden reaktiv aufgrund externer Anstöße oder proaktiv durch selbsttätiges Identifizieren von Bedarfslücken bzw. Opportunitäten entwickelt werden.“ (Martin et al. 2008)*

Die Definition von Martin et al. mit ihren zusätzlichen Merkmalen des Crowdsourcing ermöglicht eine klare Abgrenzung gegenüber verwandten Konzepten.

Doan et al. bezeichnen ein System als Crowdsourcing-System, wenn es eine Menge von Personen anwirbt, sich an der Lösung eines Problems zu beteiligen, das der System-Betreiber definiert hat:

*“We say that a system is a CS system if it enlists a crowd of humans to help solve a problem defined by the system owners.” (Doan et al. 2011, S. 87)*

Dazu definieren Doan et al. vier Herausforderungen, die beim Einsatz eines Crowdsourcing-Systems zu bewältigen sind:

- Wie können Benutzer gewonnen und gehalten werden?
- Wie können sich die einzelnen Benutzer beteiligen?
- Wie werden die Beiträge der einzelnen Benutzer zur Gesamtlösung zusammengeführt?
- Wie werden die Benutzer und deren Beiträge evaluiert?

Im Rahmen der Klassifizierung von CS-Systemen berücksichtigen Doan et al. neun Dimensionen. Zwei entscheidende Dimensionen sind die Art der Kollaboration (implizit oder explizit) sowie der Typ des Zielproblems. Explizite Systeme sind Standalone-Systeme, in denen Benutzer explizit kollaborieren. Im Kontext von Web-CS-Systemen identifizierten Doan et. al. dazu folgende unterschiedliche Ausprägungen:

- *Evaluating*: Unterstützt die Bewertung und Empfehlung von Elementen.
- *Sharing*: Unterstützt das Teilen von Elementen.
- *Networking*: Unterstützt die Vernetzung von Benutzern.
- *Building Artifacts*: Unterstützt das Erstellen von Artefakten.
- *Executing Tasks*: Unterstützt die Ausführung von Aufgaben.

Implizite Systeme unterstützen die Kollaboration zwischen Benutzern zur Lösung von Problemen in impliziter Form. Doan et al. unterscheiden bei dieser Art von Systemen zwischen *Standalone* und *Piggyback*. Im Gegensatz zu Standalone bezeichnet Piggyback die automatisierte Nutzung externer System-Aktivitäten für die Lösung der eigenen Aufgabe. Beispielsweise können die Informationen über Online-Käufe innerhalb eines

anderen Systems für das eigene System zur Generierung von Empfehlungen genutzt werden. Piggyback-Systeme müssen sich folglich nicht mit dem Anwerben von Benutzern und deren Aufgabenverteilung beschäftigen.

Um das Maß des manuellen Aufwands sowie die Rolle der realen Benutzer zu bewerten, werden sog. *Slaves*, *Perspective providers*, *Content providers* und *Component providers* unterschieden: Slaves übernehmen Teilaufgaben im Sinne des Divide-And-Conquer-Prinzips. Perspective providers bringen unterschiedliche Perspektiven ein, die zusammengefasst eine bessere Lösung ermöglichen als eine einzelne Perspektive eines einzelnen Benutzers. Content providers erzeugen eigenen Inhalt in Form von Artefakten; in der Rolle der Component providers sind die Benutzer selber eine Komponente des Ziel-Artefakts, wie beispielsweise ein Benutzer innerhalb eines sozialen Netzwerkes.

Tabelle 5 zeigt eine Übersicht über die unterschiedlichen Typen von Crowdsourcing-Systemen und charakterisiert sie in Bezug auf die zu unterscheidenden Aspekte.

Nature of Collaboration	Architecture	Must recruit users?	What users do?	Examples	Target Problems	Comments
Explicit	Standalone	Yes	Evaluating ► review, vote, tag	► reviewing and voting at Amazon, tagging Web pages at del.icio.us.com and Google Co-op	Evaluating a collection of items (e.g., products, users)	Humans as perspective providers. No or loose combination of inputs.
			Sharing ► items ► textual knowledge ► structured knowledge	► Napster, YouTube, Flickr, CPA N, programmableweb.com ► Mailing lists, Yahoo! Answers, Q UIQ, ehow.com, Quora ► Swivel, Many Eyes, Google Fusion Tables, Google Base, bmr.b.wisc.edu, galaxyzoo, Piazza, Orchestra	Building a (distributed or central) collection of items that can be shared among users.	Humans as content providers. No or loose combination of inputs.
			Networking	► LinkedIn, MySpace, Facebook	Building social networks	Humans as component providers. Loose combination of inputs.
			Building artifacts ► software ► textual knowledge bases ► structured knowledge bases ► systems ► others	► Linux, Apache, Hadoop ► Wikipedia, openmind, Intellipedia, ecolcommunity ► Wikipedia infoboxes/DBpedia, IWP, Google Fusion Tables, YAGO- NAGA, Cimple/DBLife ► Wikia Search, mahalo, Freebase, Eureka ► newspaper at Digg.com, Second Life	Building physical artifacts	Humans can play all roles. Typically tight combination of inputs. Some systems ask both humans and machines to contribute.
			Task execution	► Finding extraterrestrials, elections, finding people, content creation (e.g., Demand Media, Associated Content)	Possibly any problem	
Implicit	Standalone	Yes	► play games with a purpose ► bet on prediction markets ► use private accounts ► solve captchas ► buy/sell/auction, play massive multiplayer games	► ESP ► intrade.com, Iowa Electronic Markets ► IM DB private accounts ► recaptcha.net ► eBay, World of Warcraft	► labeling images ► predicting events ► rating movies ► digitizing written text ► building a user community (for purposes such as charging fees, advertising)	Humans can play all roles. Input combination can be loose or tight.
	Piggyback on another system	No	► keyword search ► buy products ► browse Web sites	► Google, Microsoft, Yahoo ► recommendation feature of Amazon ► adaptive Web sites (e.g., Yahoo! front page)	► spelling correction, epidemic prediction ► recommending products ► reorganizing a Web site for better access	Humans can play all roles. Input combination can be loose or tight.

**Tabelle 5: Crowdsourcing-Systeme im WWW**

Quelle: (Doan et al. 2011, S. 88)

Die beschriebenen Klassifikationsmerkmale und Dimensionen werden in Abschnitt 5.5 genutzt, um die unterschiedlichen Aspekte der Beteiligung innerhalb von PERCOP zu konzipieren.

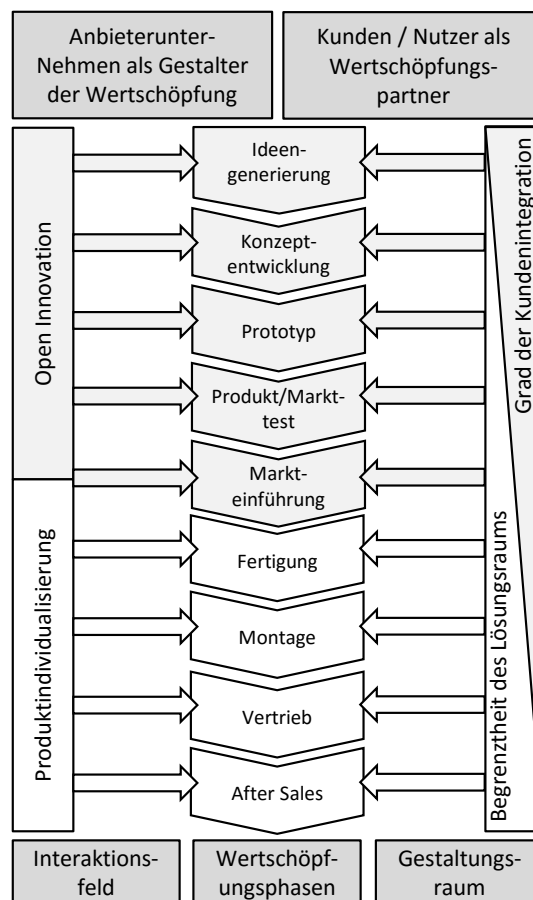
### 5.3 Interaktive Wertschöpfung

Im Zusammenhang mit Crowdsourcing fällt häufig der Begriff der *Interaktiven Wertschöpfung*. Tatsächlich existieren Gemeinsamkeiten der beiden Konzepte, allerdings besitzen die beiden Ansätze unterschiedliche Schwerpunkte und Zielsetzungen, weswegen eine synonyme Verwendung der Begriffe prinzipiell nur unter bestimmten Konstellationen und Rahmenbedingungen angebracht ist (vgl. Martin et al. 2008).

Nach Reichwald und Piller (vgl. Abbildung 48) handelt es sich bei der interaktiven Wertschöpfung um *„einen Prozess der kooperativen (und freiwilligen) Zusammenarbeit zwischen Hersteller und Kunde (Nutzer) zwischen den Extremen einer gänzlich hersteller- bzw. gänzlich kundendominierten Wertschöpfung. Die Zusammenarbeit kann sich sowohl auf operative Aktivitäten als auch auf eine Produkt- und Prozessentwicklung beziehen. Der interaktive Wertschöpfungsprozess wird dabei entweder durch das Unternehmen oder durch den Kunden initiiert“* (Reichwald und Piller 2006, S. 44).

Wie die Definition verdeutlicht, liegen bei der interaktiven Wertschöpfung die beiden Rollen *Kunde* und *Hersteller* im Fokus der Betrachtung. Ziel ist die gemeinsame Problemlösung und der soziale Austausch in Form eines freiwilligen Innovationsprozesses. Dabei tritt der Hersteller als Gestalter der Wertschöpfung auf und der Kunde als Wertschöpfungspartner.

Neben den unterschiedlichen Phasen der Wertschöpfung, angefangen bei der Ideengenerierung bis hin zur After-Sales-Phase, existieren zwei Interaktionsfelder, die die beiden unterschiedlichen Formen der interaktiven Wertschöpfung repräsentieren, *Mass Customization* und *Open Innovation*.



**Abbildung 48: Modell der interaktiven Wertschöpfung**

Quelle: (Reichwald und Piller 2006, S. 44)

### Mass Customization

Als Kombination der Begriffe *Massenproduktion* und *Individualisierung* bezeichnet *Mass Customization* ein Konzept der Produktion, bei der sowohl Eigenschaften der Massenfertigung als auch die Möglichkeit der Individualisierung der Produkte durch den Kunden kombiniert werden. Kunden werden in den Produktionsprozess eingebunden und erhalten trotz individueller Anpassungen ein Produkt, welches zu Kosten der Massenproduktion gefertigt wird.

„Produktindividualisierung (Mass Customization) ist hingegen die Zusammenarbeit zwischen Unternehmen und Kunden, die sich auf Wertschöpfungsaktivitäten im operativen Produktionsprozess bezieht und auf die Entwicklung eines individualisierten Produktes für einen Abnehmer abzielt. Ziel ist, durch Kundenintegration Zugang zu Bedürfnisinformation zu bekommen, um so die genauen Wünsche einzelner Abnehmer in einem heterogenen Markt besser erfüllen zu können.“ (Reichwald und Piller 2009, S. 53)

Beispielsweise bieten Automobil- und Computerhersteller den Kunden Konfigurationsmöglichkeiten der angebotenen Fahrzeuge bzw. Computersysteme. So können die Kunden ein gewünschtes System (in eingeschränktem Umfang) individuell anpassen.

Im Vergleich der unterschiedlichen Crowdsourcing-Ansätze weist die sog. *User-designbasierte Massenfertigung*<sup>42</sup> Übereinstimmungen mit der Mass Customization auf. Ein wesentlicher Unterschied besteht jedoch dahingehend, *„dass Produkte in Crowdsourcing-Modellen ohne die Userarbeit nicht marktfähig wären und in der Regel auch keine standardisierten Vorkonfigurationen existieren“* (Papsdorf 2009, S. 58). Generell ist Mass Customization *„auf einen konkreten Konsummarkt beschränkt, der wirtschaftliche Vorteil ist für das Unternehmen nicht frei verfügbar und es findet kein offener Aufruf statt“* (Papsdorf 2009, S. 58).

Des Weiteren ist bei Crowdsourcing-Aktivitäten die aktive Person nicht zwingend auch der Kunde des Produktes. Die Beteiligung kann andere Gründe haben; darauf wird im weiteren Verlauf der Arbeit detailliert eingegangen.

### *Open Innovation*

*Open Innovation* bezieht sich auf den Innovationsprozess zur Entwicklung neuer Produkte in großen Stückzahlen. Open Innovation beginnt bei der Ideengenerierung („dort“ ist der Grad der Kundenintegration am höchsten und nimmt von dort an stetig ab) und endet bei der Markteinführung.

Hauptziel von Open Innovation ist, *„dass zum einen durch die aktive Integration von Kunden und Nutzern in alle Phasen des Innovationsprozesses Bedürfnisinformation besser erhoben werden kann als durch klassische Maßnahmen der Marktforschung oder eines Trendscoutings. Zum anderen soll durch die Nutzung eines großen heterogenen Netzwerks an externen Experten die Lösungssuche verbessert werden“* (Reichwald und Piller 2009, S. 53).

Gassmann und Enkel zerlegen Open Innovation in die folgenden drei Kernprozesse (vgl. Gassmann und Enkel 2006, 134ff.):

- *Outside-In-Prozess*: Prozess zur Integration externen Wissens in den Innovationsprozess, beispielsweise von Lieferanten, Kunden und externen Partnern.

---

<sup>42</sup> Userdesignbasierte Massenfertigung: *„Die industrielle (Massen-) Fertigung von Produkten wird mit einer Teilindividualisierung kombiniert. Bei diesem Modus der Userereinbindung entwickeln User aus „Rohlingen“ (etwa T-Shirts, Aufkleber oder auch Tapeten) mittels bereitgestellter Online-Konfiguratoren und durch selbst kreierte Designs (Motive, Vektor-Grafiken, Druckvorlagen, CAD-Files oder Schnittmuster) eigenständige Produkte.“* (Papsdorf 2009, S. 58)

- *Inside-Out-Prozess*: Prozess zur Externalisierung internen Wissens, beispielsweise durch den Verkauf von Lizenzen oder Patenten.
- *Coupled-Prozess*: Prozess als Mischform aus Outside-In-Prozess und Inside-Out-Prozess, beispielsweise zum Setzen von Standards oder zur Gründung von Allianzen.

Abbildung 48 verdeutlicht, dass die Phase der Fertigung bereits der Produkt-individualisierung zuzuordnen ist. Wie bereits beschrieben, bezieht sich diese Produktindividualisierung auf die Wertschöpfungstätigkeiten im operativen Produktionsprozess für individuelle Produkte in Einzelanfertigung. Das Modell verdeutlicht auch, dass von der Fertigung bis zur After-Sales-Phase die Begrenztheit des Lösungsraumes stetig zunimmt.

## 5.4 Open Source

Auch der Begriff *Open Source* steht mit kollektiver Intelligenz bzw. Crowdsourcing in Beziehung. Bei Open Source liegt der Schwerpunkt jedoch vor allen Dingen auf der Erstellung von Software; Open-Source-Software (OSS) ermöglicht „*die Vereinigung zwischen Wettbewerb und kollektivem Eigentum; die Integration des Produktnutzers in Erstellungsprozess; die Beteiligung von einzelnen Programmierern ebenso wie das Engagement von großen Konzernen; die gleichzeitige Verfolgung von Wettbewerbs und Kooperation sowie die Vereinigung von altruistischem Verhalten mit ökonomischem Nutzenkalkül*“ (Brügge 2004, S. 19).

Die Open-Source-Initiative (OSI) zertifiziert Open-Source-Software anhand Ihrer selbst definierten Open-Source-Definition. Für die Zertifizierung muss eine Software folgende Eigenschaften besitzen (vgl. Open Source Initiative (OSI) 2007):

- freie Weitergabe,
- verfügbarer Quellcode,
- abgeleitete Arbeiten,
- Integrität des Autoren-Quellcodes,
- keine Diskriminierungen von Personen oder Gruppen,
- keine Einschränkung bzgl. der Nutzung,
- Lizenzerteilung,
- die Lizenz darf nicht produktspezifisch sein,
- die Lizenz darf andere Software nicht einschränken und
- die Lizenz muss technologieneutral sein.

Brügge nennt drei zusätzliche Besonderheiten von Open-Source-Software, die eine solche Software aber nicht zwingend besitzen muss (Brügge 2004, S. 21–22):

- nicht kommerzielle Einstellungen<sup>43</sup>,
- ein hoher Grad an Kollaboration<sup>44</sup> und
- starke räumliche Verteilung der Entwickler<sup>45</sup>.

Die beschriebenen Eigenschaften erfordern besondere Rahmenbedingungen bei der Durchführung von Open-Source-Software-Projekten. Organisatorisch lassen sich die Rahmenbedingungen innerhalb der beiden Kategorien *Koordination* und *Motivation* einordnen (siehe Tabelle 6).

Organisatorische Rahmenbedingungen von Open-Source-Software-Projekten	
Koordinationsbedingungen	Motivationsbedingungen
Eindeutige Schnittstellen zwischen den Programmteilen	OSS-Beteiligung stiftet Nutzen für das Erreichen eigener Ziele:  a) Reputationsmöglichkeit bei geschätzter Gruppe in Abhängigkeit von OSS-Beitragshöhe b) Programmieren macht Spaß und man erweitert seine Fähigkeiten c) OSI-Lizensierung führt zu verbessertem Programm und Marktchancen, die der Entwickler benötigt
Klare Regeln für die Übermittlung von Programmcode	
Innovation und Weiterentwicklung durch inkrementelle Schritte möglich	
Hohe Transparenz	
Weltweite Vernetzung und Kommunikation zu geringen Kosten	
Von der Community anerkannter Projektleiter und starke Implementierung der Antiforking-Norm	Lizenzen, Maintainerlisten und Namensschutz verhindern, dass der Programmierer um die Früchte seiner Arbeit betrogen wird
Niedrige Eintrittsbarrieren und großer Entwicklerpool	
Ausreichende Ressourcen	

**Tabelle 6: Organisatorische Rahmenbedingungen von OSS-Projekten**

Quelle: (Brügge 2004, S. 62)

<sup>43</sup> Ursprünglich entstand die Idee und Umsetzung von Open-Source-Software zwar mit nicht-kommerziellem Interesse; durch die Entstehung von Distributoren sind jedoch auch Anbieter mit kommerziellem Interesse hinzugekommen (vgl. Brügge 2004, S. 21).

<sup>44</sup> Grad und Art der Arbeitsteilung in den Projekten können stark variieren, von Gruppen gleichberechtigter Beteiligten bis hin zu Konstellationen ohne Weisungsbefugnisse und Hierarchien (vgl. Brügge 2004, S. 22).

<sup>45</sup> Auch der Grad der geographischen Verteilung kann variieren; beispielsweise kann Open Source Software ausschließlich an einem Ort entstehen (vgl. Brügge 2004, S. 22).

Bezüglich der Motivation lassen sich Gemeinsamkeiten zu der in Abschnitt 5.1 beschriebenen kollektiven Intelligenz erkennen. So entspricht der Aspekt *Love* dem in Tabelle 6 formulierten Ziel (b), *Glory* entspricht Ziel (a) und *Money* lässt sich Ziel (c) zuordnen.

## 5.5 Potenzial der Crowd für das PERCOP-Framework

Die vorherigen Abschnitte haben gezeigt, dass die Intelligenz der Masse in verschiedenen Bereichen eingesetzt und in diversifizierter Art und Weise genutzt wird. Insbesondere Crowdsourcing besitzt Eigenschaften, die den entsprechenden Anforderungen des PERCOP-Frameworks nahekommen. Daher werden im Folgenden den verschiedenen Aufgabenblöcken des PERCOP-Frameworks geeignete Möglichkeiten der Beteiligung zugeordnet.

Tabelle 7 zeigt das Ergebnis der Zuordnung. Die Tabelle ist im Anhang der Arbeit noch einmal mit einem größeren Schriftgrad zu finden.

Die Tabelle baut in ihrer Grundstruktur auf der Darstellung von Doan et al. (vgl. Doan et al. 2011) zur Charakterisierung von Crowdsourcing-Systeme im WWW auf (vgl. Tabelle 5), berücksichtigt aber zudem die weiteren relevanten Dimensionen. Für jede Aktivität in PERCOP (siehe oben Abschnitt 3.2.2) enthält die Tabelle eine entsprechende Zeile. In der Zeile werden einer Aktivität jeweils die möglichen Arten der Kollaboration (implizit und/oder explizit) zugeordnet. Zudem werden die relevanten Eigenschaften spezifiziert; relevante Eigenschaften sind für die PERCOP-Aktivitäten

- das adressierte Problem entsprechend der Klassifizierung von Crowsourcing-Systemen nach Doan et al. (siehe oben Tabelle 5: Target Problems),
- die Ausprägung des CS-Systems ebenfalls nach Doan et al. (Evaluating, Sharing etc., siehe oben),
- eine Spezifikation der Aktivität unter Einbeziehung der Ausprägung und
- die Rolle des realen Benutzers ebenfalls nach Doan et al. (Slaves, verschiedene Typen von Providern, siehe oben).

Schließlich werden die vier von Doan et al. identifizierten wesentlichen Herausforderungen bei Crowdsourcing-Systemen als Kernfragen für die einzelnen Aktivitäten beantwortet. Die Einträge der Tabelle werden nachfolgend erläutert.



PERCOP Beteiligung	Art der Kollaboration		Eigenschaften				Kernfragen			
	Explizit	Implizit	Adressiertes Problem	Ausprägung des CS-Systems	Tätigkeiten	Benutzer-Rolle	Wie können Benutzer gewonnen und gehalten werden?	Wie können sich die einzelnen Benutzer beteiligen?	Wie werden die Beiträge der einzelnen Benutzer zur Gesamtlösung zusammengeführt?	Wie werden die Benutzer und deren Beiträge evaluiert?
Umgebungen und Funktionalitäten definieren	X		Building a collection of items	Sharing Evaluating	Defining Environments and their functionalities	content provider perspective provider	ask for volunteers (pay/require users) + instant gratification	Web-Portal: Umgebungen und Funktionen definieren	Umgebungen und deren Funktionalitäten dienen den anderen Aktivitäten als Basis	Kontrolle durch CS-Betreiber
Regeln erstellen	X	(X)	Building a collection of items	Sharing Evaluating	Creation, sharing and evaluation of rules	content provider perspective provider	ask for volunteers (pay/require users)	Web-Portal: Regeln über Assistent manuell erfassen, implizit über Aufzeichnung von Aktivitäten	Regeln verbinden bestehende Elemente (Sensordaten, Funktionen, Informationen) miteinander. Qualitätskontrolle erfolgt aktiv über Benutzer.	Kontrolle durch "Umgebungs-Ersteller"
Funktionalitäten zuordnen	X		Building a collection of items	Sharing Evaluating	Creation, sharing and evaluation of mappings	content provider perspective provider	ask for volunteers (pay/require users)	Web-Portal: Zuordnung von Aktionen und Informationen zu Funktionalitäten	Funktionalitäten werden direkt mit Informationsartefakten verknüpft. Qualitätskontrolle erfolgt aktiv über Benutzer.	Kontrolle durch andere Benutzer implizite Kontrolle durch Bewertungen
Informationen zuordnen	X		Building physical artifacts	Sharing Evaluating	Creation, sharing and evaluation of mappings	content provider perspective provider	ask for volunteers (pay/require users)	Web-Portal: Zuordnung von Aktionen und Informationen zu Funktionalitäten	Informationen werden direkt mit Funktionalitäten verknüpft. Qualitätskontrolle erfolgt aktiv über Benutzer.	Kontrolle durch andere Benutzer implizite Kontrolle durch Bewertungen
Informationsartefakte erstellen	X		Building physical artifacts	Sharing	Creation of information artifacts	content provider	ask for volunteers (pay/require users)	Erstellen von Informationsartefakten innerhalb beliebiger (externer) Systeme	Bestehende Informationsartefakte werden durch Verknüpfung integriert.	Kontrolle durch externe Systeme, bei denen die Artefakte liegen
Informationsartefakte semantisch anreichern	X	X	Evaluating (tag artifacts)	Evaluating (tag artifacts)	Manually/automatically tagging Rating of "Answers"	all roles possible	ask for volunteers pay for service	Ergänzung von Metadaten zu bestehenden Artefakten	Semantische Anreicherung erleichtert situationsgerechtes Auffinden von Artefakten	Kontrolle durch externe Systeme, bei denen die Artefakte liegen
Sensoren entwickeln	X		Building physical artifacts	Building physical artifacts	Software development	content provider	ask for volunteers (pay/require users) + ownership situations	Entwicklung von Software-Komponenten in Form von Open-Source-Projekten	Die Sensoren übertragen ihre ermittelten Werte an den Information Broker und können dort weiterverwendet werden	Kontrolle durch andere Entwickler
Recommender-Module entwickeln	X		Building physical artifacts	Building physical artifacts	Software development	content provider	ask for volunteers (pay/require users) + ownership situations	Entwicklung von Software-Komponenten in Form von Open-Source-Projekten	Recommender-Module sind in der Lage, aus Input-Daten eine Liste von (relevanten) Artefakten zu ermitteln	Kontrolle durch andere Entwickler
Recommender-Konfigurationen definieren	X	X	Building physical artifacts	Building physical artifacts	Creation of Configurations	all roles possible	ask for volunteers pay for service	Erstellung von spezifischen Empfehlungs-Konfigurationen	Recommender-Konfigurationen kombinieren verschiedene Empfehlungsmodule miteinander	Kontrolle durch andere Benutzer implizite Kontrolle durch Bewertungen
Informationsvorschläge bewerten	X	X	Evaluating	Evaluating	Rating of information artifacts	perspective provider (explicit)	ask for volunteers pay for service	Bewertung von erhaltenen Informationsartefakten (geeignet/nicht geeignet)	Bewertungsinformationen werden innerhalb der Informationssuche berücksichtigt	Kontrolle durch die Anwendung und das Benutzerprofil
Recommender-Konfigurationen bewerten		X	Evaluating	Evaluating	Rating of recommender configurations	perspective provider (implicit)		Bewertung von Recommender-Konfigurationen erfolgt implizit über die Bewertung von Informationen	Bewertungsinformationen werden innerhalb der Informationssuche berücksichtigt	Kontrolle durch die Anwendung
Allgemeine Beteiligung	X	X	Kontextabhängige Informationsversorgung in Echtzeit	Kontextabhängige Informationsversorgung in Echtzeit	see pc-contributions	all roles possible	ask for volunteers pay users require users pay for service	Unterschiedliche Beteiligungsformen	Siehe PERCOP-Übersicht	Unterschiedliche Kontrollmechanismen

Tabelle 7: Analyse der Beteiligung im Rahmen von PERCOP<sup>46</sup>

Bezüglich der Kollaborationsart lässt sich feststellen, dass die Gesamtheit der Beteiligungsaktivitäten innerhalb des Frameworks explizit ist; einige Aktivitäten werden jedoch ergänzt durch implizite Kollaboration. Beispielsweise wird die explizite Aktivität „*Informationsvorschläge bewerten*“ durch implizite Aktivitäten ergänzt. So kann z. B. die Bewertung eines Informationsvorschlags gleichzeitig genutzt werden, um die eingesetzte Empfehlungskonfiguration dynamisch anzupassen. Auch die Anreicherung von Informationsartefakten kann implizit erfolgen sowie auch die Erstellung von Regeln, da diese automatisiert beispielsweise durch das Aufzeichnen von Benutzeraktivitäten erfolgen kann.

Die adressierten Probleme beziehen sich vor allen Dingen auf das Erstellen von Artefakten (*building physical artifacts*), wie z. B. die Entwicklung von Software (Sensoren und Empfehlungsmodule) oder die Erstellung von Regeln, Informationsartefakten oder Zuordnungen. Die Artefakte können auch kollaborativ erstellt und/oder kollaborativ

<sup>46</sup> Die Einträge der Tabelle sind bewusst sprachlich gemischt (Deutsch/English), um den Bezug zu der Übersicht von Doan et al. (vgl. Tabelle 5) zu verdeutlichen.

bewertet werden (*sharing, evaluating*). Die Aktivitäten, bei denen Informationsartefakte semantisch angereichert werden, adressieren das Problem *tag artifacts*. Die Bewertung der Informationsvorschläge entspricht dem *evaluating*-Problem.

Bezüglich der Benutzerrollen werden für die einzelnen Aktivitäten äquivalent zu den adressierten Problemen und Tätigkeiten *content provider* und *perspective providers* benötigt. Die Benutzerrollen *slaves* und *component providers* sind weniger relevant, können jedoch bei den Definitionen der Empfehlungskomponenten oder der semantischen Anreicherung der Informationsartefakte zum Einsatz kommen.

In Hinblick auf die Kernfragen können folgende Eigenschaften des Frameworks identifiziert werden:

#### *Wie können Benutzer gewonnen und gehalten werden?*

Die Gewinnung neuer Benutzer kann prinzipiell in allen Beteiligungsbereichen in Form eines offenen Aufrufs (*ask for volunteers*) oder auf Anweisung (*require users*) bzw. gegen eine Belohnung (*pay users*) erfolgen. Ziel von PERCOP ist es, zunächst die Idee des offenen Aufrufs zu realisieren, um einen großen Nutzen durch einen hohen Verbreitungsgrad zu erreichen. Durch die generische Realisierung ergeben sich aber auch Anwendungsszenarios, bei denen beispielsweise im Rahmen einer unternehmensinternen Nutzung oder durch die Integration von Herstellern die beiden anderen Formen sinnvoll sein können. Auf diese Aspekte wird im Rahmen der Fallbeispiele näher eingegangen (vgl. Kapitel 9).

Auch die Option *pay for service* kann innerhalb von PERCOP genutzt werden, beispielsweise bei der Bewertung von Informationsvorschlägen: Einem Benutzer könnten nur dann Vorschläge angezeigt bekommen, wenn er auch in regelmäßigen Abständen Vorschläge in Hinblick auf ihre Relevanz bewertet. Ein analoges Verfahren kann bei der Anreicherung der Informationsartefakte mit semantischen Informationen eingesetzt werden.

Um Benutzer zu „halten“ und dazu zu bringen, sich dauerhaft zu beteiligen, können weitere Mechanismen eingesetzt werden. Beispielsweise führt *instant gratification* dazu, dass Benutzer das Gefühl haben, etwas Sinnvolles zu tun, da sie ständig darüber informiert werden, in welcher Form ihre Beteiligung sich auf das System auswirkt. Bei der Erstellung von Sensoren und Empfehlungsmodulen kann es sinnvoll sein, dem Benutzer das Gefühl zu geben, dass er ein Teil des Systems „besitzt“ (*ownership situation*) (vgl. Doan et al. 2011, S. 94); dadurch fühlt er sich eventuell verpflichtet, sich auch weiterhin um seinen Anteil zu kümmern. Anderen Verfahren, wie beispielsweise die Integration von Spielen

(*gamification*), wird auf den ersten Blick möglicherweise eine geringe der Bedeutung zugeschrieben. Sie sind jedoch auch denkbar, um die Beteiligung zu fördern.

#### *Wie können sich die einzelnen Benutzer beteiligen?*

Auf die Frage nach der konkreten Beteiligung der Benutzer wurde bereits detailliert eingegangen. Auch die Antwort auf die Frage nach der Zusammenführung der Einzelbeteiligungen zu einer Gesamtlösung ergibt sich aus den vorhergehenden Erläuterungen.

#### *Wie werden Benutzer und ihre Beiträge evaluiert?*

Die Auswahl einer Umgebung und deren Funktionalitäten dienen als Basis für alle weiteren Aktivitäten innerhalb des PERCOP-Frameworks. Daher ist es sinnvoll, dass eine übergeordnete Instanz (z. B. der CS-Plattformbetreiber) diese Aktivität evaluiert. Die erstellten Regeln werden sinnvollerweise wiederum durch den „Umgebungs-Ersteller“ überwacht. Die Zuordnungen der Informationen bzw. der Funktionalitäten kann zum einen durch andere Benutzer evaluiert werden, zum anderen in Form einer impliziten Kontrolle durch die Bewertungen.

Da die Erstellung der Informationsartefakte kein Bestandteil des Frameworks ist, sondern innerhalb externer Systeme geschieht, können auch die zugehörigen Kontrollaktivitäten nur dort stattfinden. Ebenso verhält es sich mit der Anreicherung der Artefakte durch semantische Daten, da diese am besten direkt mit den Artefakten verknüpft werden. Denkbar wäre allerdings auch eine externe Kontrolle durch Community-Mitglieder möglicherweise ergänzt durch Automatismen, die die Autoren bei dem Prozess der Anreicherung (*Tagging*) unterstützen (Siehe Kapitel 7).

Da die Aktivitäten der Sensor- und Empfehlungsmodul-Entwicklung in Form von Open-Source-Projekten durchgeführt werden (können), erfolgt die Evaluation durch die anderen Entwickler. Werden interne Module entwickelt (beispielsweise für die explizite Nutzung innerhalb einer Organisation), muss die Kontrolle durch das organisationsinterne Entwicklungsteam erfolgen. Die Definition der Empfehlungskonfigurationen wird zum einen explizit durch andere Benutzer, zum anderen implizit durch die Bewertungen evaluiert.

Die Bewertung der Informationsvorschläge erfolgt entsprechend der Konzeption von PERCOP explizit durch den jeweiligen Benutzer. Die Evaluation erfolgt durch die Anwendung bzw. durch das Benutzerprofil. So kann beispielsweise über die Gesamtheit

aller Bewertungen eines Informationsvorschlags in Abhängigkeit der jeweiligen Benutzerprofile die Empfehlungen für zukünftige Benutzer angepasst werden.

## 5.6 Analyse der Motivation zur Beteiligung

Die Ausführungen im letzten Abschnitt haben die Beteiligung durch die Crowd bei den verschiedenen Aktivitäten im PERCOP-Framework analysiert. Die folgenden Abschnitte widmen sich den Anreizen, die zur Beteiligung motivieren können. Dazu werden verschiedene Arten der Motivation analysiert und im Kontext des Frameworks diskutiert.

Eine einheitliche Motivationstheorie existiert nicht, vielmehr wird Motivation in der Philosophie, der Psychologie, der Wirtschaftspsychologie und in der Managementlehre betrachtet, so dass sich Quellen in den zugehörigen Veröffentlichungsorganen finden. Klassifiziert werden können die Theorien zur Motivation zudem abhängig davon, ob sie auf die Identifikation von Bedürfnissen (Inhaltstheorien) oder auf kognitive Prozesse (Prozesstheorien) ausgerichtet sind.

### 5.6.1 Inhaltstheorien zur Motivation

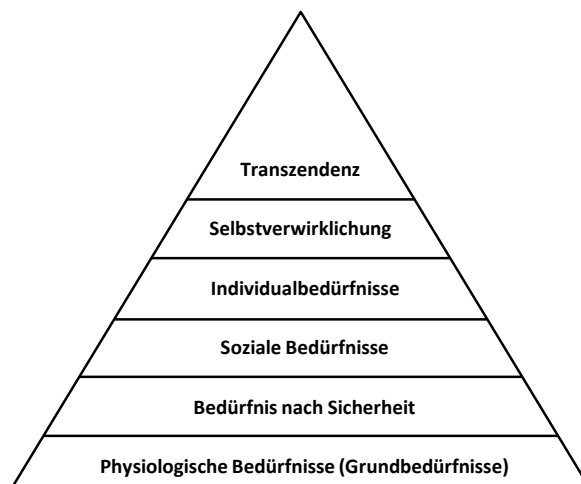
Inhaltstheorien erklären, was in einer Person oder in seiner Umwelt Verhalten bzw. Handeln erzeugt und aufrechterhält, wobei der Fokus in erster Linie auf konkreten Bedürfnissen und Motiven liegt. Somit ist das Ziel von Inhaltstheorien die Identifikation von Bedürfnissen, die durch menschliches Handeln befriedigt werden sollen.

Die folgenden Theorien sind bekannte Vertreter der Inhaltstheorien; sie werden im Folgenden kurz erläutert:

- die Bedürfnispyramide von Maslow,
- die ERG-Theorie von Alderfer,
- die Zwei-Faktoren-Theorie von Herzberg und
- die Leistungsmotivationstheorie von McClelland et al.

#### 5.6.1.1 Die Bedürfnispyramide

Als Ausgangspunkt für die Untersuchung der menschlichen Motivation wird in der Literatur häufig auf die Bedürfnispyramide von Maslow verwiesen (vgl. Maslow 1981). Diese Pyramide stellt unterschiedliche Bedürfnisarten hierarchisch auf ihren Ebenen dar, wobei „der Mensch“ die Befriedigung seiner Bedürfnisse von der unteren Schicht beginnend nach oben anstrebt (vgl. Abbildung 49).



**Abbildung 49: Maslow's (erweiterte) Bedürfnispyramide**

Quelle: In Anlehnung an (Maslow 1981; Maslow 1943; Maslow 1971)

Auf der untersten Ebene sind die physiologischen Bedürfnisse angeordnet. Dazu zählen beispielsweise Essen, Trinken und Schlafen. Eine Ebene höher befinden sich die Bedürfnisse nach Sicherheit (z. B. materielle und berufliche Sicherheit). Auf der nächsten Ebene strebt der Mensch nach sozialen Bedürfnissen und Bedürfnissen der Zuneigung (z. B. Liebe und Freundschaft). Ebene vier beschreibt die Bedürfnisse nach Achtung und Wertschätzung. Die fünfte Ebene schließlich repräsentiert die Selbstverwirklichung. Ergänzt wurde das Modell später durch eine zusätzliche Ebene, der Transzendenz, welche das Streben nach Gott bzw. einer vergleichbaren Dimension abbildet (vgl. Maslow 1971). Es ist nicht zwingend erforderlich, dass ein Bedürfnis vollständig befriedigt ist, bevor das nächst höher gelegene Bedürfnis eintritt.

Obwohl das Modell von Maslow in der Forschung häufig als Ausgangsbasis genutzt wird, sieht es sich gleichwohl häufig auch der Kritik ausgesetzt; insbesondere wird dem Modell vorgeworfen, dass eine hinreichende Absicherung durch empirische Untersuchungen fehlt. Zudem wird die Rangfolge der Bedürfnisse in Frage gestellt, was auch durch empirische Untersuchungen zur Überprüfung des Modells bestätigt wurde (vgl. Gebert und Rosenstiel 2002).

#### **5.6.1.2 Die ERG-Theorie**

Die sog. ERG-Theorie bildet eine Weiterentwicklung der Bedürfnispyramide durch Alderfer (vgl. Alderfer 1972). Er definiert die Bedürfnisklassen *Existence* (Existenz), *Relatedness* (Beziehung) und *Growth* (Wachstum): Existenzbedürfnisse umfassen beispielsweise Sicherheit, Gehalt und Arbeitsbedingungen. Beziehungsbedürfnisse

werden durch soziale Bedürfnisse wie Achtung und Wertschätzung repräsentiert. Wachstumsbedürfnisse sind z. B. Selbstverwirklichung, Produktivität und Entfaltung.

Auf Basis empirischer Untersuchung stellt Alderfer folgende Hypothesen zur Motivation auf (vgl. Alderfer 1969):

- *Frustrations-Hypothese*: Ein nicht befriedigtes Bedürfnis bleibt/wird dominant.
- *Frustrations-Regressions-Hypothese*: Wird ein Bedürfnis nicht befriedigt, steigt ein niedrigeres/früheres Bedürfnis auf und wird dominant.
- *Befriedigungs-Progressions-Hypothese*: Wird ein Bedürfnis befriedigt, aktiviert dies ein anderes (höheres) Bedürfnis<sup>47</sup>.
- *Frustrations-Progressions-Hypothese*: Auch Misserfolg kann ein Beitrag zur Selbstverwirklichung und zu Persönlichkeitswachstum darstellen.

Aus den Thesen leitet Alderfer die folgenden sieben Aussagen (*Propositions*) ab (vgl. Alderfer 1969):

- Je weniger die Existenzbedürfnisse befriedigt sind, desto stärker werden sie.
- Je weniger die Beziehungsbedürfnisse befriedigt sind, desto stärker werden die Existenzbedürfnisse.
- Je mehr die Existenzbedürfnisse befriedigt sind, desto stärker werden die Beziehungsbedürfnisse.
- Je weniger die Beziehungsbedürfnisse befriedigt sind, desto stärker werden sie.
- Je weniger die Wachstumsbedürfnisse befriedigt sind, desto stärker werden die Beziehungsbedürfnisse.
- Je mehr die Beziehungsbedürfnisse befriedigt sind, desto stärker werden die Wachstumsbedürfnisse.
- Je mehr die Wachstumsbedürfnisse befriedigt sind, desto stärker werden sie.

Somit reduziert Alderfer zum einen die Anzahl der Ebenen der Maslow'schen Bedürfnispyramide. Zum anderen verlässt er die strikte Ordnung der Hierarchie.

### 5.6.1.3 Die Zwei-Faktoren-Theorie

Herzberg's Zwei-Faktoren-Theorie repräsentiert eine Inhaltstheorie speziell zur Arbeitsmotivation (vgl. Herzberg et al. 1959). Dabei werden Zufriedenheit und Unzufriedenheit als zwei unabhängige Dimensionen betrachtet, die durch sog. *Motivatoren* und *Hygienefaktoren* repräsentiert werden. Motivatoren (wie beispielsweise

---

<sup>47</sup> Diese These liegt der Maslow'schen Bedürfnispyramide zu Grunde.

Anerkennung, Erfolg und Verantwortung) beeinflussen als intrinsische Faktoren (siehe Abschnitt 5.6.3) die Zufriedenheit; sie entstehen aus dem Arbeitsinhalt.

Hygienefaktoren hingegen beziehen sich auf die Arbeitsumwelt, repräsentieren somit extrinsische Faktoren, welche Einfluss auf die Unzufriedenheit ausüben. Beispiele für Hygienefaktoren sind Gehalt, Beziehung zu Mitarbeitern und Arbeitsbedingungen.

Aus der unterschiedlichen Betrachtung von Zufriedenheit und Unzufriedenheit Innerhalb der Zwei-Faktoren-Theorie ergibt sich, dass „*unzufrieden*“ und „*nicht zufrieden*“ bzw. „*zufrieden*“ und „*nicht unzufrieden*“ nicht dieselbe Bedeutung besitzen. Sind beispielsweise alle Hygienefaktoren positiv ausgeprägt, entsteht keine Zufriedenheit, sondern lediglich eine „*Nicht-Unzufriedenheit*“.

Somit kommen Herzberg et al. zu dem Schluss, dass nur die intrinsischen Faktoren (Motivatoren) zu einer Zufriedenheit führen und Motivation erzeugen.

#### 5.6.1.4 Die Theorie der Leistungsmotivation

Basierend auf der Persönlichkeitstheorie von Murray (vgl. Murray 1938) entwickelten McClelland et al. eine Theorie, die insbesondere die *Leistung* als Motiv betrachtet (vgl. McClelland et al. 1953). McClelland et al. definieren drei wesentliche Schlüsselbedürfnisse:

- das Bedürfnis nach Erfolg (Need for Achievement) als *Leistungsmotiv*,
- das Bedürfnis nach Zugehörigkeit (Need for Affiliation) als *Anschlussmotiv* und
- das Bedürfnis nach Macht (Need for Power) als *Machtmotiv*.

Eine Person mit starkem Leistungsmotiv bevorzugt Aufgaben und Situationen, „*bei denen sie durch ihre Fähigkeiten und ihren Einsatz erfolgreich sein kann*“ (Hentze 2005, S. 121); zu solchen Fähigkeiten zählen beispielsweise Selbstvertrauen und Eigeninitiative.

Besitzt eine Person ein hohes Anschlussmotiv, stehen Akzeptanz, Beliebtheit und Harmonie im Vordergrund. Die Person strebt somit eine positive soziale Beziehung an.

Ein ausgeprägtes Machtmotiv beschreibt Personen, die eine Beeinflussung der Verhaltensweisen anderer anstreben. Sie haben das Bedürfnis und den Wunsch nach Status und Aufstieg (vgl. Hentze 2005, S. 121).

## 5.6.2 Prozesstheorien zur Motivation

Im Gegensatz zu den Inhaltstheorien befassen sich die Prozesstheorien mit den kognitiven (psychischen) Prozessen, die dazu führen, dass Personen Entscheidungen für ein Handlungsziel treffen. Somit steht der Prozess, der einen Menschen dazu führt, eine bestimmte Alternative zu wählen, im Fokus. Bekannte Ansätze für Prozesstheorien sind<sup>48</sup>:

- die VIE-Theorie von Vroom,
- die Leistungsmotivation von Atkinson,
- die Gerechtigkeitstheorie von Adams und
- die Attributionstheorie von Weiner.

### 5.6.2.1 Die VIE-Theorie

Die Valenz-Instrumentalitäts-Erwartungs-Theorie beschreibt als mathematisch formulierte Funktion eine Errechnbarkeit der Bedeutung des Handelns (vgl. Vroom 1964). Die wesentlichen zugehörigen Variablen sind Valenz, Instrumentalität und Erwartung: *Valenz* bezeichnet „die Stärke der Bevorzugung bestimmter Objekte oder Handlungen für das Individuum gegenüber dem erreichbaren Ziel oder Endergebnis“ (Hentze et al. 2001, S. 549). Ihr Wert liegt zwischen +1 (positive Ergebnisse) und -1 (negative Ergebnisse). Den Wert 0 besitzen neutrale Ergebnisse.

Die *Instrumentalität* repräsentiert einen Schätzwert der subjektiven Wahrscheinlichkeit für das Eintreten eines Ereignisses. Der Wert liegt zwischen +1 (das Ereignis tritt ein bzw. das Ziel wird durch den Handlungsausgang erreicht) und -1 (das Ereignis tritt nicht ein bzw. das Ziel wird durch den Handlungsausgang nicht erreicht).

Unter *Erwartung* wird die Wahrscheinlichkeit des konkreten Handlungsausgangs verstanden. Der Wert liegt zwischen 0 (unwahrscheinlich) und 1 (wahrscheinlich). Die Entscheidung bzw. die Motivationsstärke eines Individuums ergibt sich durch die Kombination aus Valenz und Erwartung:

$$\text{Entscheidung} = \text{Valenz}(E) * \text{Erwartung}(E)$$

Nach Vroom wird eine Person nur dann zu einer bestimmten Leistung motiviert, wenn diese den höchsten Entscheidungswert besitzt.

---

<sup>48</sup> Weitere Ansätze sind: *Erwartungs-Wert-Modell* von Porter und Lawler (vgl. Staehle et al. 1999, 237ff.), die *Zieltheorie* von Locke und Latham (vgl. Staehle et al. 1999, 236ff.)



### 5.6.2.2 Leistungsmotivation

Die Leistungsmotivation nach Atkinson beschreibt die Handlungstendenz eines (emotionalen) Konflikts zwischen Erfolg/Hoffnung und Misserfolg/Furcht (vgl. Atkinson und Birch 1978).

$$T_e = M_e * W_e * A_e$$

Die Tendenz auf Streben nach Erfolg ( $T_e$ ) wird bestimmt durch Motiv ( $M_e$ ), subjektive Erfolgswahrscheinlichkeit ( $W_e$ ) und subjektivem Anreiz des Erfolges ( $A_e$ ). Äquivalent dazu definiert Atkinson die Tendenz zur Vermeidung von Misserfolg als Ergebnis der Kombination aus Motiv zur Vermeidung von Misserfolg, der subjektiven Wahrscheinlichkeit und des Anreizes von Misserfolg. Die resultierende Tendenz ergibt sich aus der Summe der Erfolgstendenz und der Misserfolgstendenz.

### 5.6.2.3 Die Gerechtigkeitstheorie

Die Gerechtigkeitstheorie bzw. Gleichgewichtstheorie nach Adams geht davon aus, dass im Kontext der Arbeitsmotivation ein Mensch zwischen sich und seinen Kollegen ein Zustand des Gleichgewichts herzustellen versucht (vgl. J. Stacy Adams 1965 und Walster et al. 1978). In Bezug auf Input und Output stellt sich die Theorie folgendermaßen dar:

$$\frac{\text{Output A}}{\text{Input A}} = \frac{\text{Output B}}{\text{Input B}}$$

Bei Vorhandensein von Ungleichgewichten streben die Mitarbeiter danach, wieder ein kognitives Gleichgewicht herzustellen. Ein Gleichgewicht kann beispielsweise durch folgende Maßnahmen erreicht werden:

- Änderung des Inputs oder des Outputs,
- verzerrte Wahrnehmung des eigenen bzw. des fremden Inputs/Outputs und
- Beeinflussung oder Wechsel der Vergleichsperson.

Im Fokus der Gerechtigkeitstheorie steht somit der Entstehungsprozess der Motivation.

### 5.6.2.4 Die Attributionstheorie

Gegenstand der Attributionstheorie von Weiner sind die Ursachen für das Zustandekommen von Erfolg und Misserfolg. Weiner ordnet die Ursachen in die drei Dimensionen Lokalisierung (intern/extern), Stabilität (stabil/variabel) und Kontrollierbarkeit (kontrollierbar/unkontrollierbar) ein (vgl. Weiner 1994): Lokalisierung bezieht sich auf die eigene (intern) oder eine andere (extern) Person. Stabilität bezieht sich auf den

Zeitverlauf und bestimmt, ob die Ursache stabil oder variabel ist. Kontrollierbarkeit repräsentiert die eigene Einflussmöglichkeit.

Des Weiteren ergeben sich nach Weiner die vier Ursachenfaktoren *Fähigkeiten*, *Anstrengung*, *Zufall* und *Schwierigkeit*. Fähigkeiten und Anstrengung repräsentieren interne Merkmale (personenbezogen), während Zufall (Glück/Pech) und Schwierigkeit externe Merkmale darstellen (situationsbezogen).

Nach Weiner (vgl. wieder Weiner 1994) werden nach Wahrnehmung eines Ereignisses drei unterschiedliche Arten der Bewertung unterschieden:

- Bewertung in Bezug auf die Erreichung des Ziels (erreicht oder nicht erreicht),
- Zurückführung auf die Ursachenfaktoren und
- Einordnung der Ursachen in die drei Dimensionen.

### 5.6.3 Intrinsische und extrinsische Motivation

Im Rahmen der Self-Determination-Theory (SDT) unterscheiden Deci und Ryan zwei unterschiedliche Formen der Motivation, die intrinsische und die extrinsische Motivation (vgl. Deci und Ryan 1985). Intrinsische Motivation bezeichnet die Bewältigung einer Aufgabe um ihrer selbst willen, ohne Fokus auf eine Belohnung oder Vorteile. Die Freude bei der Bearbeitung der Aufgabe stellt den wesentlichen Motivationsaspekt dar.

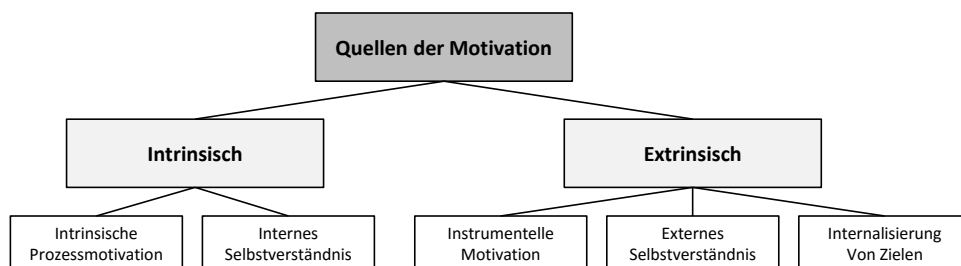
*“Intrinsic motivation is defined as the doing of an activity for its inherent satisfactions rather than for some separable consequence. When intrinsically motivated a person is moved to act for the fun or challenge entailed rather than because of external prods, pressures, or rewards.”*  
(Ryan und Deci 2000, S. 56)

Ist eine Person extrinsisch motiviert, steht der eigene Vorteil durch die Erbringung einer Leistung im Mittelpunkt.

*“Extrinsic motivation is a construct that pertains whenever an activity is done in order to attain some separable outcome.”* (Ryan und Deci 2000, S. 60)

Nach Barbuto und Scholl lassen sich den beiden Motivationstypen fünf Quellen der Motivation zuordnen (vgl. Barbuto und Scholl 1998). Quellen für intrinsische Motivation sind danach die intrinsische Prozessmotivation und das interne Selbstverständnis: Der Spaß an der Arbeit als alleinige Motivation für deren Durchführung bildet die intrinsische Prozessmotivation. Internes Selbstverständnis bezieht sich auf den Fall, dass ein Individuum innengelenkt (engl. *inner-directed*) agiert, beispielsweise durch individuelle Werte, Charakterzüge und Kompetenzen.

Extrinsische Motivation basiert nach Barbuto und Scholl auf instrumenteller Motivation, externem Selbstverständnis und der Internalisierung von Zielen. Instrumentelle Motivation bezeichnet die Motivation aufgrund von externen (materiellen) Einflüssen wie Bezahlung oder Boni. Bei externem Selbstverständnis agiert das Individuum fremdbestimmt aufgrund von Anforderungen aus der Umwelt oder eines Teams. Bei der Internalisierung von Zielen bildet der Wunsch des Erreichens eines gemeinsamen Ziels die Quelle der Motivation. Abbildung 50 visualisiert die fünf Quellen der Motivation nach Barbuto und Scholl.

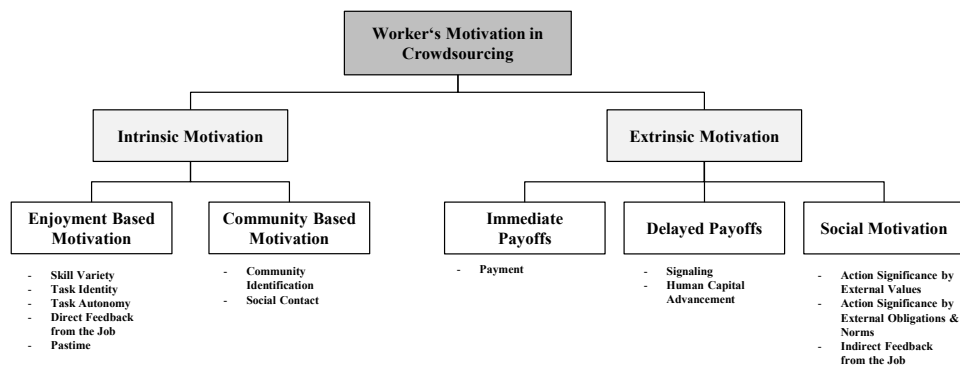


**Abbildung 50: Fünf Quellen der Motivation**

Quelle: In Anlehnung an (Barbuto und Scholl 1998)

Eine ähnliche Klassifizierung findet sich bei Kaufmann et al. (vgl. Kaufmann et al. 2011). Mit Bezug auf Lakhani und Wolf (vgl. Lakhani und Wolf 2005), Hackmann und Oldham (vgl. Hackman und Oldham 1980) und Weiss (vgl. Weiss 1995), erstellen Kaufmann et al. ein Modell für die Motivation innerhalb des Crowdsourcing, indem die intrinsische und die extrinsische Motivation genauer klassifiziert werden (siehe Abbildung 51). Das Modell diene zur konkreten Untersuchung von Motivationsaspekten, hier am Beispiel einer Studie zu *Mechanical Turk*<sup>49</sup>.

<sup>49</sup> Bei *Mechanical Turk* handelt es sich um eine Plattform, auf der Kunden einfache Arbeitsaufgaben an die Community auslagern können. Weitere Informationen finden sich unter: <https://www.mturk.com/mturk/>



**Abbildung 51: A Model for Worker's Motivation in Crowdsourcing**

Quelle: (Kaufmann et al. 2011)

Für die intrinsische Motivation werden zwei Kategorien unterschieden, *Enjoyment Based Motivation* und *Community Based Motivation*. *Enjoyment Based Motivation* bezieht sich auf den Spaß der Personen bei der Durchführung einer Aufgabe; dieser Spaß kann durch folgende Aspekte bewertet/„gemessen“ werden (vgl. Kaufmann et al. 2011, S. 4):

- *Skill Variety*: Verwendung der Vielfältigkeit von Fähigkeiten, die benötigt werden, um eine bestimmte Aufgabe zu erfüllen und die den Fähigkeiten der Person entsprechen. Je höher die Auswahl der passenden Fähigkeiten, desto größer sollte die Motivation sein, eine spezifische Aufgabe zu wählen.
- *Task Identity*: Bezieht sich darauf, wie hoch die Wahrnehmung der Erfüllung einer Aufgabe durch den Arbeiter ist. Je näher das Ergebnis der Arbeit heranrückt, desto größer ist die Motivation.
- *Task Autonomy*: Bezieht sich auf die Freiheiten, die dem Arbeiter bei der Erledigung seiner Aufgabe gegeben werden. Die Motivation ist höher, je mehr eigene Entscheidungen und Kreativität erlaubt sind.
- *Direct Feedback from the Job*: Bezieht sich auf das Maß der Wahrnehmung eines Gefühls für Erfolg während oder nach der Ausführung einer Aufgabe, explizit eingeschränkt auf direktes Feedback durch die Arbeit, nicht durch andere Personen.
- *Pastime*: Bezieht sich auf die Beteiligung zum Zeitvertreib. Entsteht, wenn die Personen etwas tun, um Langeweile zu vermeiden.

Bei der zweiten Kategorie, der Kategorie *Community Based Motivation*, wird die Person durch eine Community-Plattform unterstützt. Folgenden Konstrukte sind in dieser Kategorie von Bedeutung:

- *Community Identification*: Bezieht sich auf das Verhalten der Benutzer, welches durch die unterbewusste Aneignung von Normen und Werten der Crowdsourcing-Plattform-Community gesteuert wird. Das Verhalten ist zurückzuführen auf einen

individuellen Identifikationsprozess (basierend auf der Idee der grundsatz-bedingten Motivation).

- *Social Contact*: Bezieht sich auf Motivation, die durch die bloße Existenz der Community erzeugt wird, welche Möglichkeit zur Förderung der sozialen Kontakte bietet.

Im Kontext der extrinsischen Motivation werden drei Kategorien der Motivation differenziert:

- *Immediate Payoffs*: Bezeichnet den direkten Erhalt einer Entschädigung, beispielsweise monetäre Vergütung für eine durchgeführte Aufgabe.
- *Delayed Payoffs*: Bezieht sich auf Vorteile, die erst im späteren Verlauf entstehen. Beispielsweise können Teilnehmer auf sich aufmerksam machen oder durch die Teilnahme die eigenen Fähigkeiten verbessern, um daraufhin eine bessere Position im Job zu erlangen.
- *Social Motivation*: Bezeichnet den äußeren (sozialen) Einfluss. Der Einfluss kann sich zum einen auf persönliche Werte einer Person beziehen. Zum anderen können auch externe Verpflichtungen oder Abhängigkeiten die Durchführung von Crowdsourcing-Aufgaben begründen.

Auf Basis der unterschiedlichen Aspekte, Betrachtungsweisen und Studien bzgl. Motivation im Rahmen des Crowdsourcing stellt Brabham bezogen auf die Frage „*Why did individuals participate in the Next Stop Design project?*“ folgende Hypothese auf:

*“Crowdsourcing participation behavior will be motivated by a mixture of intrinsic and extrinsic motivators, as well as rational, norm-based, and affective motivators, including the opportunity for career advancement; collaborative contribution; peer recognition; self-expression; learning of new skills; and the prospect of having fun.” (Brabham 2012, S. 316)*

Seine Untersuchung hat die Hypothese bestätigt, hat zudem ermittelt, dass unterschiedliche Antreiber die Motivation zur Beteiligung bei den Teilnehmern bilden (vgl. Tabelle 8).

Da die Untersuchung für ein konkretes Projekt durchgeführt wurde, kann sie nicht als allgemeingültig für Crowdsourcing-Projekte angesehen werden. Die Beteiligungsmöglichkeiten innerhalb des in dieser Arbeit vorgestellten Frameworks sind in einigen Bereichen vergleichbar, jedoch deutlich vielfältiger. Daher kann davon ausgegangen werden, dass auch innerhalb von PERCOP unterschiedliche Gründe für eine Beteiligung eine Rolle spielen.

Motivation	Kategorie
Vorantreiben der eigenen Karriere	Extrinsisch, rational
Von anderen Teilnehmern wahrgenommen werden	Extrinsisch, rational, normenbasiert
Teilhaben an einer gemeinsamen (kollaborativen) Aufgabe	Intrinsisch und extrinsisch, normenbasiert
Sich selbst ausdrücken	Intrinsisch, emotional
Spaß haben	Intrinsisch, emotional
Neue Fähigkeiten und Kenntnisse aneignen	Intrinsisch und extrinsisch, rational
Geringe Eintrittsbarriere, Einfachheit der Nutzung, ansprechende Wahrnehmung, nützliche Webseite	Extrinsisch, rational

**Tabelle 8: Zusammenfassung der Erkenntnisse zur Motivation bei Next Stop Design<sup>50</sup>**

Quelle: In Anlehnung (Brabham 2012, S. 320)

Im Folgenden werden diese in Hinblick auf das Framework untersucht, dabei werden die zuvor erläuterten Aspekte der Motivation auf PERCOP übertragen.

### 5.6.4 Motivationspotenzial innerhalb von PERCOP

Für die Beteiligungsmöglichkeiten innerhalb von PERCOP existieren verschiedene Anreize. Solche Beteiligungsmöglichkeiten sind:

- Umgebungen und Funktionalitäten auswählen,
- Regeln erstellen,
- Funktionalitäten zuordnen,
- Informationen zuordnen,
- Informationsartefakte erstellen,
- Informationsartefakte semantisch anreichern,
- Sensoren entwickeln,
- Empfehlungsmodule entwickeln,
- Empfehlungskonfigurationen definieren und
- Informationsvorschläge bewerten.

<sup>50</sup> Weitere Informationen siehe: <http://dbrabham.wordpress.com/crowdsourcing/next-stop-design/>

Die einzelnen Tätigkeiten können jeweils unabhängig voneinander und von unterschiedlichen Individuen durchgeführt werden. Die Tätigkeiten werden im Folgenden in Hinblick auf die zuvor beschriebenen Motivationstypen und Klassifikationen untersucht.

Die initiale Aktivität innerhalb des Frameworks ist die Auswahl einer Umgebung; eine Umgebung stellt bestimmte Funktionalitäten bereit. Aufgrund der „generischen“ Realisierung des Frameworks können Umgebungen unterschiedlichen Typs sein, beispielsweise Anwendungsprogramme oder eine Kombination aus Anwendungsprogrammen.

Die Motivation zur Auswahl einer Umgebung kann grundsätzlich intrinsisch oder extrinsisch sein. Da jedoch die Auswahl einer Umgebung keine Aktivität in diesem Sinne darstellt, werden insbesondere extrinsische Gründe vorliegen. Solche Gründe sind beispielsweise das eigene Interesse an einer Informationsbeschaffung innerhalb der definierten Umgebung oder die Bereitstellung eigener Informationen für andere Benutzer.

Regeln verbinden eine Situation bzw. Kontexte (beispielsweise bestimmt durch Aktionen von Benutzern oder durch den Zustandswechsel ihrer jeweiligen Umgebung) mit konkreten Informationen oder Funktionalitäten. Der Komplexitätsgrad von Regeln kann unterschiedlich sein. Regeln verwenden bestimmte Elemente (Sensoren, Sensordaten, Funktionalitäten und Informationen) und schaffen eine Verbindung zwischen diesen Elementen. Daher wird für die Erstellung von Regeln kein technisches Wissen benötigt. Die Motivation für die Erstellung von Regeln kann sowohl extrinsisch als auch intrinsisch ausgeprägt sein. Zum einen kann der eigene Vorteil im Mittelpunkt stehen oder die externe Beeinflussung (extrinsisch), zum anderen können auch persönliche Werte oder der Spaß an der Aufgabe die Beteiligung bei der Regelerstellung begründen (intrinsisch).

Ähnlich zur Erstellung von Regeln verhält es sich bei der Zuordnung von Funktionalitäten und Informationen. Die Trennung der Zuordnung ermöglicht eine Betrachtung auf unterschiedlichen Abstraktionsebenen. Individuen können durch Regeln bestimmen, welche Funktionalität aktuell ausgeführt wird. An einer anderen Stelle (und möglicherweise durch andere Individuen) wurden diesen Funktionalitäten Informationen zugeordnet. Auch die Motivation variiert. So können beispielsweise Hersteller die Funktionen einer Anwendung definieren und Regeln formulieren, wann diese Funktionalitäten ausgeführt bzw. genutzt werden, damit zielgerichtet passende Informationen angezeigt werden können. So können bei den Herstellern die Gründe extrinsischer Art sein, während bei dem Individuum, welches die Informationen zuordnet, intrinsische Gründe überwiegen.

Die semantische Anreicherung von Informationen verhält sich ähnlich in Bezug auf die Motivation. Die Gründe für das Erstellen eines Dokuments können beispielsweise intrinsischer Art sein, während die Anreicherung des Dokumentes die Auffindbarkeit erhöhen soll und damit extrinsische Gründe repräsentiert.

Bei der Entwicklung von Sensoren und Empfehlungsmodulen können Motivationserkenntnisse aus dem Bereich der Open-Source-Software herangezogen werden, da es sich auch hierbei um konkrete softwaretechnische Lösungen handelt. In Hinblick auf die Motivation in Open-Source-Projekten haben verschiedene Studien gezeigt, dass der Spaß und die Freude bei der Entwicklung von Software und der Lösung von Problemen Individuen motivieren, sich zu beteiligen und zu engagieren. Des Weiteren spielen die eigene Reputation sowie das gemeinsame Erreichen eines Ziels innerhalb eines Teams eine große Rolle (vgl. Brabham 2010, Hars und Ou 2002, Hertel et al. 2003 sowie Bonaccorsi und Rossi 2003).

Empfehlungskonfigurationen stellen eine Kombination von Empfehlungsmodulen für einen bestimmten Kontext bzw. für eine konkrete Situation innerhalb einer spezifischen Umgebung dar. Sie entscheiden, welche Quelle genutzt wird, und damit auch über die Qualität der ermittelten Informationsartefakte. Die Konfigurationen werden zum einen implizit über die Bewertungen der Informationsvorschläge erstellt bzw. ausgewählt. Zum anderen können die Konfigurationen von Benutzern explizit definiert werden. Gründe für die Beteiligung können auch bei dieser Aktivität intrinsischer und extrinsischer Art sein.

Zusammenfassend stellt Tabelle 9 die Beteiligungsmöglichkeiten im Kontext des Frameworks dar und liefert Beispiele zu den jeweiligen Motivationsausprägungen. Die Übersicht besitzt keinen Anspruch auf Vollständigkeit, sondern soll „nur“ mögliche Motivationsaspekte<sup>51</sup> verdeutlichen.

Die Übersicht verdeutlicht, dass unterschiedliche Motivationsaspekte bei der Nutzung von PERCOP existieren. Somit ist das Ziel für die Erstellung des Frameworks in Hinblick auf die Beteiligung der einzelnen Individuen, die unterschiedlichen Motivationsaspekte der einzelnen Aktivitäten bestmöglich zu unterstützen bzw. zu fördern.

---

<sup>51</sup> Beispielsweise könnte die intrinsische Motivation „Spaß an der Durchführung“ für alle Tätigkeiten in Frage kommen, ebenso wie die extrinsische Motivation „monetäre Vergütung“. Jedoch zeigen bestehende Konzepte wie beispielsweise Wikis und Open-Source-Software, dass bestimmte Motivationsaspekte eine höhere Relevanz besitzen als andere.



Beteiligungsmöglichkeit	Intrinsische Motivation	Extrinsische Motivation
Umgebungen auswählen		<ul style="list-style-type: none"> <li>Einfache Nutzung</li> <li>Eigenes Interesse an Unterstützung</li> <li>Hersteller-Interesse</li> </ul>
Regeln erstellen	<ul style="list-style-type: none"> <li>Spaß an der Tätigkeit, logische Verknüpfungen zu erstellen</li> </ul>	<ul style="list-style-type: none"> <li>Eigener Vorteil bei Zuordnung eigener Dokumente</li> </ul>
Funktionalitäten zuordnen Informationen zuordnen	<ul style="list-style-type: none"> <li>Gefühl, Teil eines Ganzen zu sein</li> </ul>	<ul style="list-style-type: none"> <li>Aufforderung von externer Seite</li> </ul>
Informationsartefakte erstellen	<ul style="list-style-type: none"> <li>Bedürfnis, Wissen oder Meinungen mitzuteilen</li> </ul>	<ul style="list-style-type: none"> <li>Von außen vorgegebene Tätigkeit</li> <li>Verbesserung der eigenen Reputation</li> </ul>
Informationsartefakte semantisch anreichern	<ul style="list-style-type: none"> <li>Bedürfnis, anderen zu helfen</li> </ul>	<ul style="list-style-type: none"> <li>Eigener Vorteil, da Dokumente besser auffindbar werden</li> </ul>
Sensoren entwickeln Empfehlungsmodule entwickeln	<ul style="list-style-type: none"> <li>Spaß an der Entwicklung und der Lösung von SE-Problemen</li> <li>Neue Fähigkeiten aneignen (aus Sicht des inneren Antriebs)</li> </ul>	<ul style="list-style-type: none"> <li>Eigener Vorteil durch die Unterstützung eigener Systeme</li> <li>Neue Fähigkeiten aneignen (in Hinblick auf die Karriere)</li> </ul>
Empfehlungskonfigurationen definieren		<ul style="list-style-type: none"> <li>Eigener Vorteil</li> <li>Monetäre Vergütung</li> </ul>
Informationsvorschläge bewerten	<ul style="list-style-type: none"> <li>Ist einfach und macht Spaß</li> </ul>	<ul style="list-style-type: none"> <li>Eigener Vorteil, da die Qualität der Informationssuche verbessert wird.</li> </ul>

**Tabelle 9: Motivationsbetrachtung innerhalb von PERCOP**

Für eine Analyse der konkreten Umsetzungslösungen unter besondere Berücksichtigung der Motivation und Beteiligung der Benutzer, werden die zehn Beteiligungs-Schnittpunkte hier in fünf Kategorien eingeteilt:

- *Software-Entwicklung*
  - Sensoren entwickeln
  - Empfehlungsmodule entwickeln
- *Autorentätigkeiten*
  - Informationsartefakte erstellen
  - Informationsartefakte semantisch anreichern
- *Bewertungen*
  - Informationsvorschläge bewerten
  - (Recommender und -Konfigurationen bewerten)

- (Sensoren bewerten)
- *Konfigurationen*
  - Regeln definierten
  - Empfehlungskonfigurationen erstellen
- *Definition und Zuordnung*
  - Umgebungen und Funktionalitäten definieren
  - Informationen zuordnen
  - Funktionalitäten zuordnen

Die Kategorie *Software-Entwicklung* beinhaltet die konkrete softwaretechnische Umsetzung von (virtuellen) Sensoren und Empfehlungsmodulen. Um die Vorteile der Crowd in dieser Kategorie bestmöglich zu nutzen, ist es sinnvoll, etablierte Lösungen für Open-Source-Softwareprojekte zu integrieren. Hier haben sich Portale wie beispielsweise SourceForge<sup>52</sup>, CodePlex<sup>53</sup>, Google Code<sup>54</sup> oder GitHub<sup>55</sup> etabliert. Diese Plattformen bieten Entwicklern Funktionalitäten zur Bereitstellung und zur kollaborativen Arbeit an Softwareprojekten. Darunter fallen Werkzeuge zur Versionsverwaltung (z. B. Subversion<sup>56</sup> oder Git<sup>57</sup>) sowie zur Dokumentation (z. B. Wikis oder Foren). Die Plattformen unterstützen dabei die intrinsischen und extrinsischen Motivationsaspekte.

Die *Autorentätigkeiten* (Erstellung der Informationsartefakte) sind nicht Bestandteil des Frameworks. Es wird davon ausgegangen, dass Informationen unterschiedlicher Art und Weise vorliegen. Die Erstellung und Verbreitung erfolgt auf den jeweiligen bestehenden Plattformen. Auch die semantische Anreicherung mit Metadaten sollte bei der Informationsquelle geschehen. Im Rahmen der vorliegenden Arbeit soll ein Annotations-Schema entwickelt werden, welches als Basis genutzt werden kann, um die Zuordnung zu konkreten Kontextinformationen zu verbessern (siehe Abschnitt 7.2.3).

Bei der Umsetzung eines Prozesses zur semantischen Annotation von Web-Informationsartefakten sollten auch Motivationsaspekte berücksichtigt werden. Beispielsweise kann durch Vorschläge ein Autor motiviert werden, sein Dokument direkt mit sinnvollen Metadaten zu annotieren. Kapitel 7 geht detailliert auf den Prozess zur Anreicherung von Web-Dokumenten mit Metadaten ein.

*Bewertungen* spielen innerhalb von PERCOP insbesondere auf der Seite der Endbenutzer eine Rolle. Der Benutzer kann direkt in der Client-Anwendung die Relevanz der erhaltenen

---

<sup>52</sup> Siehe <http://www.sourceforge.net>

<sup>53</sup> Siehe <http://www.codeplex.com>

<sup>54</sup> Siehe <http://code.google.com>

<sup>55</sup> Siehe <http://github.com>

<sup>56</sup> Siehe <http://subversion.apache.org/>

<sup>57</sup> Siehe <http://git-scm.com/>

Information bewerten. Dazu gilt es, ein sinnvolles User-Interface zur Bewertung bereitzustellen. So könnte beispielsweise über eine Skalenbewertung der Relevanzgrad durch Benutzer bestimmt werden. Im Rahmen der Konzeption des Frameworks und des Prototyps wird jedoch eine binäre Bewertung eingesetzt. Somit kann der Benutzer mit nur einer Aktion die Information als „*relevant*“ oder „*nicht relevant*“ kennzeichnen. Die Einfachheit erhöht die Bereitschaft der Benutzer, eine Bewertung abzugeben. Eine Überlegung, zu wie viel Prozent die Information relevant zum aktuellen Kontext ist, könnte sich negativ auf die Beteiligungsbereitschaft auswirken. Dieser Aspekt überwiegt den des Mehrwerts einer Skalenbewertung für das Framework.

Für die Aktivitäten innerhalb der Kategorien *Konfigurationen* und *Definition/Zuordnung* bietet sich ein zentrales Web-Portal an, auf dem die Crowd kollaborativ und transparent agieren kann. Das Portal sollte die ermittelten Motivationspotentiale zielgerichtet unterstützen. Der Aufbau einer Community unterstützt beispielsweise die Motivation in Hinblick auf Stolz und Reputation, indem andere Teilnehmer sehen, wie sich die einzelnen Individuen beteiligen. Des Weiteren kann extrinsische Motivation gefördert werden, indem beispielsweise monetäre Vergütungsfunktionalitäten integriert werden.

Im Rahmen der Prototypentwicklung (siehe Kapitel 8) soll die Förderung der Motivation innerhalb von PERCOP noch einmal diskutiert werden.

## 6 Empfehlungssysteme und -verfahren

In Systeme, in denen Objekte erworben, konsumiert oder genutzt werden, sind heute zu meist Empfehlungssysteme integriert. Dies gilt insbesondere für den Online-Vertrieb von Produkten, Musik oder Filmen. Dieser Ansatz lässt sich auf das „Konsumieren“ von Informationsartefakten übertragen, da auch hier die Meinung anderer über das jeweilige Artefakt hilfreich sein kann. In PERCOP sind daher Empfehlungssysteme integriert, nicht zuletzt, um die Menge irrelevanter Informationen zu reduzieren.

In diesem Kapitel wird dazu zunächst allgemein auf die Charakteristika von Empfehlungssystemen eingegangen (Abschnitte 6.1-6.2). Anschließend werden konkrete Verfahren und Algorithmen von Empfehlungssystemen erläutert und in Hinblick auf ihre Nutzbarkeit in PERCOP analysiert (Abschnitte 6.3-6.6). Unter Berücksichtigung der erarbeiteten Ergebnisse wird in Kapitel 6.7 ein Empfehlungssystem entwickelt, das die Anforderungen an ein Empfehlungssystem für PERCOP erfüllt.

### 6.1 Charakteristika von Empfehlungssystemen

Für den Begriff Empfehlungssystem<sup>58</sup> (engl. *Recommender System*) finden sich in der Literatur zahlreiche Definitionen; sie fokussieren aufgrund differenzierter Betrachtungsweisen unterschiedliche Aspekte.

Ursprünglich wurden Empfehlungssysteme definiert als Systeme, in denen Personen Empfehlungen als Input erzeugen, die das System aggregiert und potenziellen Empfängern vermittelt (vgl. Burke 2002 und Resnick und Varian 1997). Im Laufe der Zeit wurden unterschiedliche Aspekte von Empfehlungssystemen und ihrer Nutzung untersucht und herausgestellt. So definieren beispielsweise Felfernig et al. ein Empfehlungssystem folgendermaßen:

*„Recommender systems support users by identifying interesting products and services in situations where the number and complexity of offers outstrips the user’s capability to survey them and reach a decision.“ (Felfernig et al. 2007, S. 1)*

Bei der Definition nach Felfernig et al. steht das Informationsüberangebot im Vordergrund; dieses Überangebot erschwert es dem Benutzer, eine Entscheidung zu treffen.

Klahold definiert ein Empfehlungssystem als „System, das einem Benutzer in einem gegebenen Kontext aus einer gegebenen Entitätsmenge aktiv eine Teilmenge „nützlicher“

---

<sup>58</sup> Die Begriffe *Empfehlungssystem*, *Kollaboratives Filtern* und *Personalisierung* werden in der Literatur häufig synonym verwendet.

*Elemente empfiehlt.*“ (Klahold 2009, S. 1) Die Definition ist sehr allgemein gefasst, und fokussiert insbesondere die Aspekte des Empfehlens nützlicher Elemente bei einer gegebenen Ausgangslage.

Renneberg berücksichtigt wesentliche einflussnehmende Aspekte und kommt dadurch zu folgender Definition:

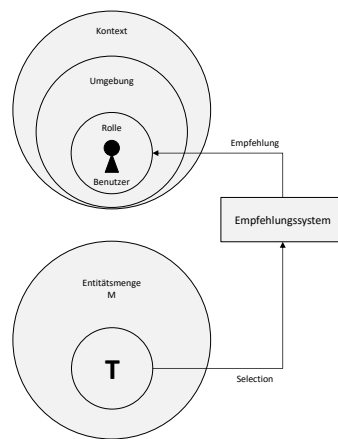
*„Ein Recommendersystem (auch Empfehlungssystem genannt) ist ein personalisiertes System, das die Selektion, Aufbereitung und Präsentation von Informationsobjekten auf die Bedürfnisse des Nutzers oder die Nutzergruppe abstimmt.“ (Renneberg 2010, S. 25)*

Die Eigenschaften von Empfehlungssystemen nach der Definition von Renneberg sind auch für PERCOP relevant, da sie den Zielen und Anforderungen des Frameworks entsprechen; allerdings fehlt der Bezug zu einem Kontext bzw. zu einer Situation (vgl. Klahold 2009). Deshalb wird dieser Arbeit eine Kombination der Definitionen von Klahold und Renneberg zugrunde gelegt:

*„Ein Empfehlungssystem ist ein System, das (innerhalb eines gegebenen Kontextes) die Selektion, Aufbereitung und Präsentation von Informationsobjekten auf die Bedürfnisse des Nutzers abstimmt“*

Die wesentlichen Bestandteile eines Empfehlungssystems in PERCOP sind somit die Benutzer und die Informationsobjekte, im folgenden Elemente genannt. Die Situation bzw. der Kontext, indem sich die Benutzer und die Elemente befinden, können in dieser Betrachtung als optional angesehen werden, da auch ohne konkreten Kontext prinzipiell eine Erzeugung von Empfehlungen möglich ist. Jedoch spielt der Kontext eine essentielle Rolle innerhalb von PERCOP, weswegen er auch in der vorliegenden Arbeit detailliert betrachtet wird.

Abbildung 52 stellt die Hauptbestandteile eines PERCOP-Empfehlungssystems und ihre Beziehungen untereinander grafisch dar.



**Abbildung 52: Empfehlungssystem innerhalb von PERCOP**

Die Aufgabe eines Empfehlungssystems besteht darin, zunächst aus einer bestimmten Menge von Elementen (M) relevante Elemente (T) auszuwählen und anschließend dem Benutzer in geeigneter Form zu präsentieren. Der Benutzer, dem Elemente empfohlen werden, kann sich in einer konkreten Situation befinden, die den Kontext bestimmt. Der Kontext ergibt sich durch den Benutzer und dessen Profileigenschaften innerhalb einer Umgebung.

## 6.2 Nutzung von Empfehlungssystemen

Personen nutzen Empfehlungssysteme mit unterschiedlicher Zielsetzung. Herlocker et al. identifizieren dazu folgende allgemeine (domänenübergreifende) Aufgabenstellungen (User Tasks) für die Nutzung von Empfehlungssystemen (vgl. Herlocker et al. 2004, S. 9):

- *Annotation in Context*<sup>59</sup>: Ursprünglich entwickelt im Rahmen des Filterns von strukturierten Diskussions-Beiträgen zur Ermittlung relevanter Inhalte. Für die Filterung werden die Elemente mit Informationen angereichert (z. B. Bewertungen anderer Benutzer), um Hilfestellungen bei der Suche bzw. bei der Auswahl zu bieten.
- *Find Good Items*<sup>60</sup>: Dem Benutzer werden Elemente in Form einer gewichteten (bewerteten) Liste vorgeschlagen. Diese Liste entspricht einer Vorauswahl durch das Empfehlungssystem. Laut Herlocker et al. nutzt der Großteil kollaborativer Empfehlungssysteme diese Form der Filterung.

<sup>59</sup> Herlocker et al. verweisen auf die Projekte *Tapestry* (vgl. Goldberg et al. 1992) und *GroupLens* (vgl. Resnick et al. 1994) in Hinblick auf die beispielhafte Implementierung von *Annotation in Context*.

<sup>60</sup> Herlocker et al. verweisen auf die Projekte *Ringo* (vgl. Shardanand und Maes 1995) und *Bellcore Video Recommender* (vgl. Hill et al. 1995) in Hinblick auf die beispielhafte Implementierung von *Find Good Items*.

Auf Basis dieser allgemeinen (domänenübergreifenden) Aufgabenstellungen beschreiben Herlocker et al. weitere spezialisierte Anwendungsfälle, welche detaillierte Gründe für die Nutzung von Recommendern darstellen (vgl. Herlocker et al. 2004, S. 9–11):

- *Find All Good Items*: Ziel ist das Auffinden aller geeigneten Elemente. In bestimmten Fällen kann es wichtig sein, nicht nur ein Element oder eine Auswahl von Elementen zu erhalten, sondern es werden alle relevanten Elemente benötigt. Beispielsweise besteht vielleicht der Bedarf, alle Dokumente zu einem bestimmten Geschäftsvorfall einzusehen.
- *Recommend Sequence*: Es wird eine Sequenz von geeigneten sortierten Elementen empfohlen. Beispielsweise werden zu Lernzwecken mehrere Informationsdokumente in einer bestimmten Reihenfolge empfohlen.
- *Just Browsing*: Benutzer mögen das eigenständige „Umsehen“, ohne ein bestimmtes Ziel zu verfolgen. Dabei spielt die Unterhaltung des Benutzers und die sich daraus ergebene User Experience eine größere Rolle als beispielsweise der Algorithmus des Empfehlungssystems.
- *Find Credible Recommender*: Um Vertrauen in ein Empfehlungssystem zu erhalten, testen Benutzer häufig die Systeme durch Vergleich der von diesen Systemen empfohlenen Elemente mit den ihnen bekannten Elementen.
- *Improve Profile*: Benutzer bewerten Elemente, da sie davon ausgehen, dass eine Bewertung ihr Profil verbessert und zudem dazu führt, dass auch sie durch Bewertungen anderer Benutzer geeignetere Elemente empfohlen bekommen.
- *Express Self*: Eine Vielzahl von Benutzern hat das Bedürfnis, die eigene Meinung zu äußern. Dabei spielt die Beeinflussung der eigenen Bewertungen (siehe „Improve Profile“) keine Rolle. Die Kommunikation mit anderen Benutzern steht im Vordergrund.
- *Help Others*: Ähnlich dem Anwendungsfall „Express Self“ verspüren Benutzer Freude, indem sie Elemente bewerten, um anderen Benutzern zu helfen, beispielsweise Entscheidungen zu vereinfachen.
- *Influence Others*: Das Ziel von Benutzern kann auch in der bewussten Beeinflussung anderer Benutzer durch die (falsche) Bewertung von Elementen bestehen. Laut Herlocker et al. stellt diese Manipulation des Empfehlungssystems ein großes Problem dar.

## 6.3 Empfehlungsverfahren und -algorithmen

Empfehlungssysteme können nach Adomavicius et al. in drei Kategorien unterteilt werden (vgl. Adomavicius et al. 2005, S. 735):

- *Kollaboratives Empfehlen*: Dem Benutzer werden Elemente empfohlen, die zuvor schon von anderen Benutzern mit ähnlichen Vorlieben und Präferenzen favorisiert wurden.
- *Inhaltsbasiertes Empfehlen*: Die Empfehlung basiert auf Elementen, die ähnlich zu Elementen sind, die der Benutzer in der Vergangenheit favorisiert hat.
- *Hybride Empfehlungssysteme*: Unterschiedliche Empfehlungsverfahren (meist kollaborativ und inhaltsbasiert) werden miteinander kombiniert.

Ergänzend identifiziert Burke drei weitere Kategorien des Empfehlens (vgl. Burke 2002, S. 2):

- *Demographisch*: Empfehlungen werden auf Basis demographischer Klassen erzeugt. Die Klassen entstehen durch die Einteilung von Benutzern in Gruppen abhängig von definierten Stereotypen (z. B. *Student*, *Vater*).
- *Nutzenbasiert*: Die Empfehlungen basieren auf Berechnungen des Nutzens von Elementen für einen bestimmten Benutzer.
- *Wissensbasiert*: Es wird mit Hilfe von Benutzerprofilen und Schlussfolgerungen auf die Präferenzen und Bedürfnisse eines Benutzers geschlossen. Aus den Präferenzen und Bedürfnissen werden dann Empfehlungen abgeleitet.

Im Hinblick auf die Nutzung innerhalb des vorgestellten Frameworks unterscheiden sich die dargestellten Empfehlungskategorien in Bezug auf ihre Bedeutung und ihre Einsatzmöglichkeiten. Daher wird im Folgenden detailliert auf die unterschiedlichen Verfahren eingegangen, um ihre Relevanz für PERCOP und die Möglichkeit der Integration in das Framework zu analysieren.

### 6.3.1 Kollaboratives Empfehlen

Kollaborative Empfehlungsverfahren sind heutzutage sehr weit verbreitet und wurden auch in der Wissenschaft ausführlich behandelt (vgl. Jannach 2011, S. 13). Bei kollaborativen Empfehlungen werden Informationen über vergangene Verhaltensweisen oder Meinungen bestehender Benutzer ermittelt und gesammelt, um daraus auf Vorlieben des aktuellen Benutzers zu schließen. Als Input dabei wird eine Matrix von Benutzern und Elementen sowie deren jeweilige Bewertungen verwendet.

Output ist zum einen eine Liste von empfohlenen Elementen. Zum anderen wird ein numerischer Wert für jedes Element zurückgegeben, der abbildet, wie sehr/stark der Benutzer das Element bevorzugt. Tabelle 10 zeigt eine Benutzer-Element-Matrix mit Bewertungen von 1 (niedrig) bis 5 (hoch). Ein „?“ repräsentiert den Fall, dass für dieses Element von dem jeweiligen Benutzer noch keine Bewertung abgegeben wurde.



	Item 1	Item 2	Item 3	Item 4	Item 5
User 1	5	3	4	4	?
User 2	3	1	2	3	3
User 3	4	3	4	3	5
User 4	3	3	1	5	4
User 5	1	5	5	2	1

**Tabelle 10: Benutzer-Element-Matrix bei kollaborativen Empfehlungen**

Quelle: (Jannach 2011, S. 14)

Insgesamt basieren kollaborative Empfehlungsansätze auf der Annahme, dass Benutzer mit gemeinsamen Interessen in der Vergangenheit auch weiterhin gemeinsame Präferenzen haben werden; die Elemente selbst werden nicht betrachtet.

Aus den Anforderungen an kollaborative Verfahren ergeben sich für die Konzeption von Empfehlungssystemen vor allen Dingen folgende Fragestellungen (vgl. Jannach 2011, S. 3):

- Wie können Benutzer identifiziert werden, die ähnlichen Präferenzen besitzen, wie der Benutzer, für den eine Empfehlung benötigt wird?
- Wie kann Ähnlichkeit gemessen werden?
- Wie wird mit neuen Benutzern umgegangen, für die noch keine historischen Informationen vorliegen?
- Wie wird vorgegangen, wenn nur wenige nutzbare Bewertungen vorliegen?
- Welche alternativen Verfahren können eingesetzt werden, um sinnvolle Empfehlungen zu erzeugen?

Zwei verbreitete Verfahren zur Bestimmung der Ähnlichkeit des kollektiven Empfehlens sind die *benutzerbasierte Nächster-Nachbar-Empfehlung* und die *elementbasierte Nächster-Nachbar-Empfehlung*.

#### *Benutzerbasierte Nächster-Nachbar-Empfehlung*

Bei diesem Verfahren werden der aktuelle Benutzer und die Benutzer-Element-Matrix (siehe Tabelle 10) als Input vorausgesetzt. Anschließend werden Benutzer mit ähnlichen Präferenzen ermittelt. Daraufhin werden für jedes Element, welches der Benutzer noch nicht kennt, Prognosen berechnet basierend auf den Bewertungen der ähnlichen Benutzer.

Um Ähnlichkeiten zwischen Benutzern (engl. *nearest neighbours*) zu ermitteln, kann die sog. *Pearson-Korrelation*<sup>61</sup> verwendet werden. Danach ergibt sich beispielsweise für die in Tabelle 10 dargestellte Ausgangssituation eine Ähnlichkeit zwischen den Benutzern User 1 und User 2 von 0.85.

Die ähnlichsten Benutzer gehen dann in die Berechnungsfunktion<sup>62</sup> zur Ermittlung der Empfehlungswerte (Kennziffer) eines Elements für einen konkreten Benutzer ein: Unter Berücksichtigung der ähnlichsten Benutzer für User 1 (User 2 und User 3) ist der berechnete Wert für Item 5 beispielsweise 4.87. Auf die gleiche Art und Weise wird für jedes Element der entsprechende Wert berechnet und daraufhin eine Liste von Empfehlungen erstellt.

Das beschriebene Verfahren ermöglicht die Berechnung einer Empfehlungskennziffer für unbekannte Elemente basierend auf den Empfehlungen anderer ähnlicher Benutzer.

#### *Elementbasierte Nächster-Nachbar-Empfehlung*

Die benutzerbasierte Empfehlung besitzt Defizite, wenn eine sehr große Anzahl von Benutzern und Elemente vorliegt (vgl. Jannach 2011, S. 16). Deshalb wurde die elementbasierte Nächste-Nachbar-Empfehlung entwickelt. Die Komplexität wird durch diesen Ansatz dahingehend reduziert, dass angenommen wird, dass Benutzer nur eine kleine Menge der verfügbaren Elemente bewerten. Es wird deshalb nicht im ersten Schritt die Ähnlichkeit zwischen Benutzern berechnet, sondern die Ähnlichkeit zwischen den Elementen. Auf Basis dieser Berechnungen werden anschließend die prognostizierten Empfehlungswerte für unbekannte Elemente ermittelt (vgl. Jannach 2011, S. 16).

Die Ähnlichkeiten zwischen Elementen kann beispielsweise mittels *Kosinus-Ähnlichkeitsmaß*<sup>63</sup> berechnet werden. Danach ergibt sich für das dargestellte Beispiel ein Ähnlichkeitswert für die Elemente Item 5 und Item 1 von 0.99.

<sup>61</sup> Pearson-Korrelation zur Ähnlichkeitsbestimmung zwischen den Benutzern a und b mit P als Menge der Elemente und  $\bar{r}_a$  als Durchschnittsbewertung von Benutzer a:

$$sim(a, b) = \frac{\sum_{p \in P} (r_{a,p} - \bar{r}_a)(r_{b,p} - \bar{r}_b)}{\sqrt{\sum_{p \in P} (r_{a,p} - \bar{r}_a)^2} \sqrt{\sum_{p \in P} (r_{b,p} - \bar{r}_b)^2}}$$

<sup>62</sup> Berechnungsfunktion zur Ermittlung von Empfehlungswerten für Element p für Benutzer a unter Berücksichtigung der nächsten Nachbarn und der durchschnittlichen Bewertung:

$$pred(a, p) = \bar{r}_a + \frac{\sum_{b \in N} sim(a, b) * (r_{b,p} - \bar{r}_b)}{\sum_{b \in N} sim(a, b)}$$

<sup>63</sup> Die Kosinus-Ähnlichkeit misst die Ähnlichkeit zweier n-dimensionaler Vektoren anhand derer Winkel zueinander. Dabei repräsentiert ein Vektor die Bewertungen verschiedener Benutzer eines Elementes. Prinzipiell sind Werte zwischen -1 und 1 möglich, aufgrund ausschließlich positiver Bewertungswerte liegt der Wert hier zwischen 0 und 1, wobei Werte

Da das einfache Kosinus-Ähnlichkeitsmaß nicht die unterschiedlichen Bewertungsweisen der Benutzer berücksichtigt, wurde eine erweiterte Funktion, das sog. angepasste<sup>64</sup> Kosinus-Ähnlichkeitsmaß definiert. Zudem kann die ursprüngliche Datenbasis für die Berechnung in der Form angepasst werden, dass als Bewertungen die Abweichungen von der Durchschnittsbewertung verwendet werden.

	Item 1	Item 2	Item 3	Item 4	Item 5
User 1	1.00	-1.00	0.00	0.00	?
User 2	0.60	-1.40	-0.40	0.60	0.60
User 3	0.20	-0.80	0.20	-0.80	1.20
User 4	-0.20	-0.20	-2.20	2.80	0.80
User 5	-1.80	2.20	2.20	-0.80	-1.80

**Tabelle 11: Angepasste Datenbasis**

Quelle: (Jannach 2011, S. 20)

Bezüglich der in Tabelle 10 beschriebenen Datenbasis ergibt sich eine angepasste Datenbasis (siehe Tabelle 11). Die Berechnung mittels angepasster Kosinus-Ähnlichkeit und der veränderten Datenbasis liefert beispielsweise den Wert 0.80 für die Ähnlichkeit zwischen den Elementen Item 5 und Item 1.

Im nächsten Schritt kann ein Empfehlungsmaß für einen Benutzer und ein konkretes Element (im Beispiel: User 1 und Item 5) bestimmt werden, indem eine gewichtete Summe für die Elemente des Benutzers (User 1) berechnet wird, die dem Element (Item 5) ähnlich sind (vgl. Jannach 2011, 20ff.).

### 6.3.2 Inhaltsbasiertes Empfehlen

Bei inhaltsbasierten Empfehlungen werden Informationen über die Elemente ermittelt und in Abhängigkeit von den Benutzerpräferenzen Empfehlungen berechnet.

---

um 1 eine hohe Ähnlichkeit aufweisen. Die Ähnlichkeit zweier Elemente (a, b) wird mit folgender Formel berechnet:

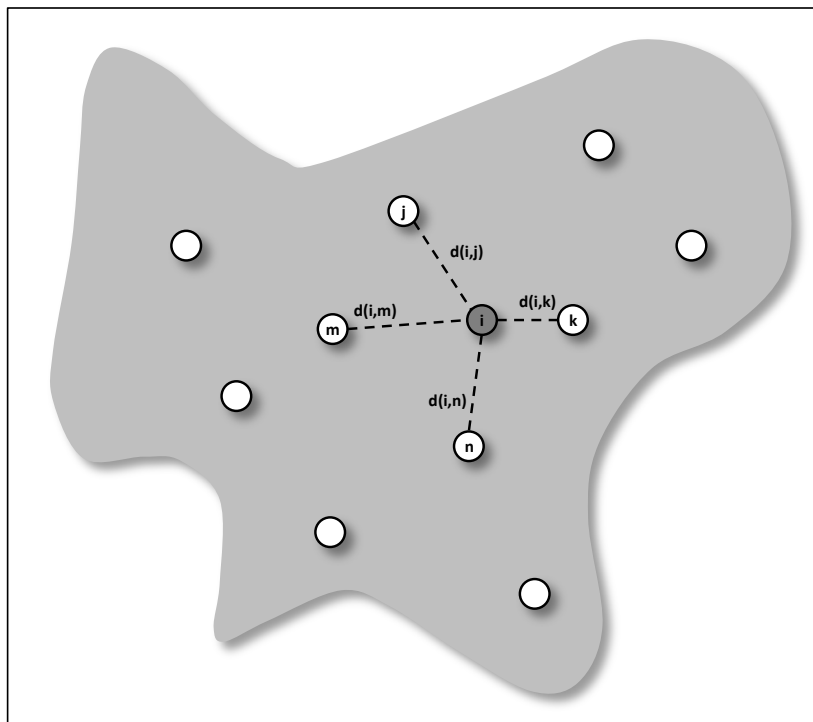
$$\text{sim}(\vec{a}, \vec{b}) = \frac{\vec{a} \cdot \vec{b}}{|\vec{a}| \cdot |\vec{b}|}$$

<sup>64</sup> Das angepasste Kosinus-Ähnlichkeitsmaß berücksichtigt auch die durchschnittliche Bewertung eines Benutzers (mit U als Menge der Benutzer, die die Elemente a und b bewertet haben) und wird berechnet mit:

$$\text{sim}(a, b) = \frac{\sum_{u \in U} (r_{u,a} - \bar{r}_u)(r_{u,b} - \bar{r}_u)}{\sqrt{\sum_{u \in U} (r_{u,a} - \bar{r}_u)^2} \sqrt{\sum_{u \in U} (r_{u,b} - \bar{r}_u)^2}}$$

Informationen über die Präferenzen oder das Verhalten anderer Benutzer werden nicht benötigt bzw. gehen nicht in die Berechnungen ein (vgl. Jannach 2011, S. 52).

Basis der inhaltsbasierten Verfahren ist die Beschreibung der Elemente in Form von Eigenschaftsinformationen<sup>65</sup>. Die Beschreibung kann manuell erstellt oder automatisch extrahiert werden. Für die Empfehlungen werden Ähnlichkeiten zwischen Elementen bestimmt; dazu kann beispielsweise der *Dice-Koeffizient* verwendet werden. Abbildung 53 visualisiert die Ähnlichkeiten zwischen Elementen: Je kürzer die Abstände zwischen zwei Elementen, desto höher ist die Ähnlichkeit.



**Abbildung 53: Ähnlichkeitsbestimmung zwischen Elementen**

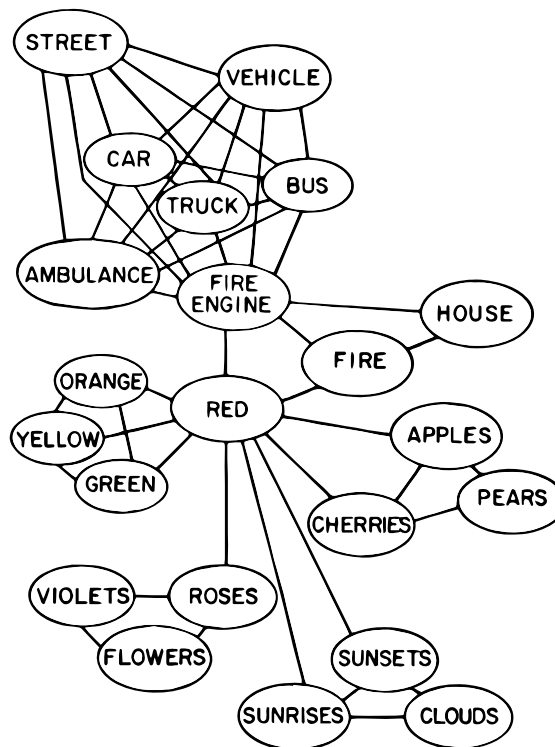
Quelle: (Celma 2010, S. 27)

In der Regel besitzen Elemente eine Vielzahl unterschiedlicher Merkmale, die für die Empfehlungen relevant sein können. Dabei kann es sich einerseits um strukturierte, andererseits um unstrukturierte Eigenschaften handeln. Aus diesem Grund wurden Ansätze entwickelt, bei denen mehrere Ähnlichkeitsmaße eingesetzt werden, um aus ihnen eine Gesamtähnlichkeit zu berechnen (vgl. Jannach 2011, S. 54). Beispiele sind unter anderem *Euclidean*, *Manhattan*, *Chebychey*, *Kosinus-Ähnlichkeitsmaß* und die *Mahalanobis-Entfernung*.

<sup>65</sup> Häufig wird auch von *Attributen* oder *Merkmalen* gesprochen.

An Stelle von einfachen Merkmalen können auch komplexere Merkmale genutzt werden, um inhaltsbasierte Empfehlungen zu erzeugen; beispielsweise können semantische Beziehungen innerhalb einer Domäne ausgenutzt werden (vgl. Hussein und Ziegler 2011). Häufig wird dazu das Verfahren der Aktivierungsausbreitung (sog. *Spreading Activation*) genutzt; bei dieser Methode werden die Elemente und deren Merkmale als Graph abgebildet, in dem die Beziehungen durch die Kanten dargestellt werden.

*„Spreading Activation bezeichnet eine Menge ähnlicher Algorithmen, die ursprünglich aus dem Bereich der künstlichen Intelligenz und der Kognitionsforschung stammen. Im Allgemeinen geht man dabei von einer bestimmten Menge von Items aus, die miteinander über Relationen verbunden sind. In diesem semantischen Netz können nun den einzelnen Items Gewichtungen zugewiesen werden, welche über die Relationen an verwandte Items propagiert werden. Dabei kann, je nach Algorithmus, der Fluss dieser Gewichtungsausbreitung durch die Stärke oder Art der Verbindung gesteuert werden und führt schließlich zu einem gewichteten Netz, in dem die am stärksten gewichteten Items als Empfehlungen genutzt werden können.“ (Hussein und Ziegler 2011, S. 3)*



**Abbildung 54: Aktivierungsausbreitung**

Quelle: (Collins und Loftus 1975, S. 412)

Im Kontext inhaltsbasierter Empfehlungen ergeben sich nach Jannach folgende Fragestellungen (Jannach 2011, S. 4):

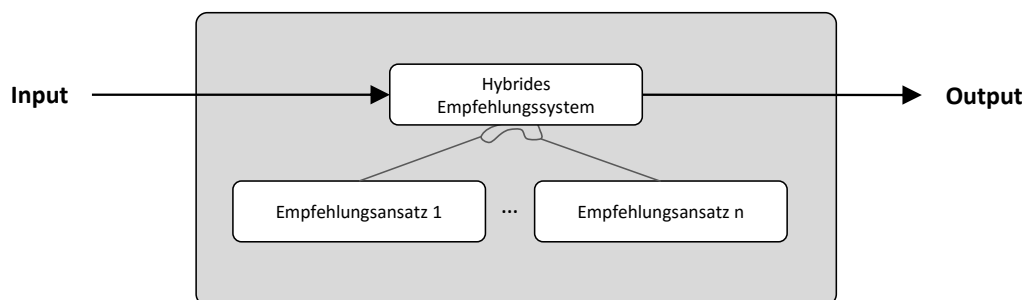
- Wie können Systeme automatisch und dauerhaft Benutzerprofile erfassen und verbessern?
- Auf welche Weise können Elemente ermittelt werden, die den Benutzerinteressen entsprechen?
- Welche Verfahren können eingesetzt werden, um automatisch Element-eigenschaften zu ermitteln und somit manuelle Benutzeraktivitäten zu reduzieren?

### 6.3.3 Hybride Ansätze

Hybride Ansätze kombinieren kollaborative und inhaltsbasierte Verfahren. Nach Jannach ergeben sich dadurch folgende Fragestellungen (vgl. Jannach 2011, S. 6):

- Welche Techniken können kombiniert werden und welche Voraussetzungen müssen für die Kombination gegeben sein?
- Sollten Vorschläge mehrerer Systeme sequenziell berechnet werden oder existieren Verfahren für eine parallele Berechnung?
- Wie sollten die unterschiedlichen Ergebnisse gewichtet werden?
- Können die Ergebnisse dynamisch erfasst werden?

Ansätze zur Kombination unterschiedlicher Methoden können in monolithische (*monolithic*), parallele (*parallelized*) oder sequenzielle (*pipelined*) Konzepte klassifiziert werden (Jannach 2011, S. 128)<sup>66</sup>. Bei der *monolithischen* Kombination von Empfehlungsansätzen werden verschiedene Ansätze zu einer Empfehlungskomponente vereint, die Input verarbeitet und daraus Output erzeugt (Abbildung 55).



**Abbildung 55: Hybrider, monolithischer Ansatz**

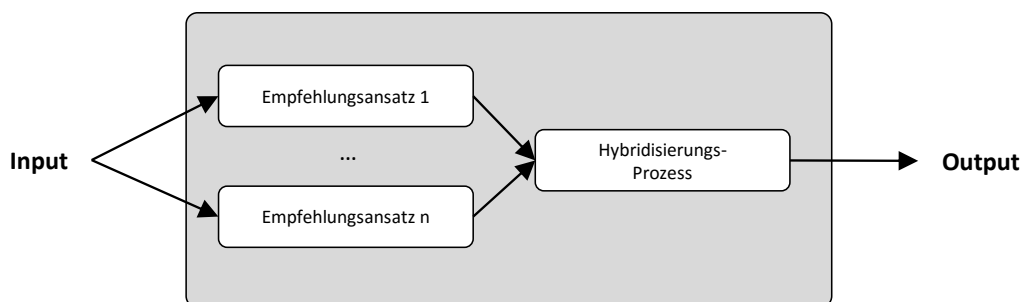
Quelle: (Jannach 2011, S. 128)

<sup>66</sup> Jannach betrachtet die von Burke (vgl. Burke 2002) definierten sieben Strategien aus einer abstrakteren Perspektive und fasst sie in 3 „Basiskonzepten“ zusammen.

Eine Umsetzung des monolithischen Ansatzes kann durch Einsatz folgender Strategien erfolgen (vgl. Burke 2007):

- *Feature Combination Strategy*: Als Ergänzung zu inhaltsbasierten Verfahren werden kollaborative Informationen hinzugezogen. Beispielsweise kann die Information über bestimmte Vorlieben eines Benutzers für ein Film-Genre eine sinnvolle Ergänzung zu der Bewertung der Benutzer (kollaborativ) und den Eigenschaften der Elemente (inhaltsbasiert) darstellen (vgl. Burke 2002, S. 8).
- *Feature Augmentation Strategy*: Im Gegensatz zur Feature Combination Strategy werden hier nicht nur verschiedene Inputtypen „einfach“ kombiniert, sondern es werden komplexere Transformationsschritte durchgeführt (vgl. Jannach 2011, S. 132). Dadurch werden Bewertungen bzw. Klassifizierungen von Elementen erzeugt, die anschließend den Input für weitere Algorithmen bilden (vgl. Burke 2002, S. 8).

Bei *parallelen* Verfahren werden zunächst mehrere Empfehlungsansätze parallel durchgeführt und anschließend durch den Prozess der Hybridisierung zu einem Output zusammengeführt (Abbildung 56).



**Abbildung 56: Hybrider, paralleler Ansatz**

Quelle: (Jannach 2011, S. 129)

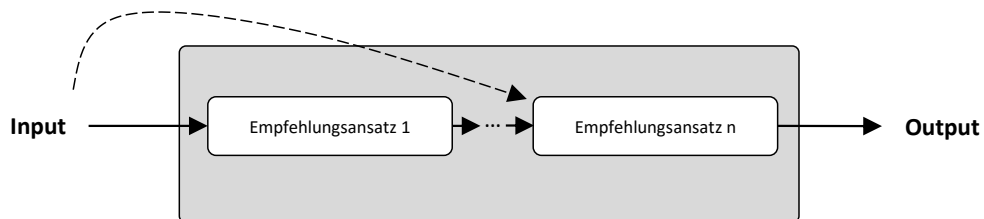
Die folgenden Strategien ermöglichen die Umsetzung des parallelen Ansatzes (vgl. Burke 2007):

- *Mixed Strategy*: Empfehlungen mehrerer Verfahren werden den Benutzern gleichzeitig präsentiert. Beispielsweise können inhaltsbasierte und kollaborative Verfahren zunächst nebeneinander ablaufen und ihre Ergebnisse anschließend zu einem Output zusammengefasst werden (vgl. Burke 2002, S. 7).
- *Weighted Strategy*: Der Empfehlungswert eines Elements wird aus allen vorhandenen Empfehlungstechniken (nach definierten Gewichtungen) berechnet. Somit können die Vorteile der einzelnen Empfehlungstechniken genutzt werden.

Durch eine dynamische Anpassung der Gewichtung können die spezifischen Eigenschaften der unterschiedlichen Techniken zur Steigerung der Qualität des Inputs berücksichtigt werden (vgl. Burke 2002, S. 6).

- *Switching Strategie*: In Abhängigkeit von der aktuellen Situation wird eine geeignete Empfehlungstechnik ausgewählt und eingesetzt. Beispielsweise können zunächst Standardtechniken genutzt werden, und nur dann, wenn diese keine sinnvollen Ergebnisse liefern, wird unter Verwendung der sog. Fallback-Strategie zu einer anderen Technik gewechselt (vgl. Burke 2002, S. 6–7).

Bei hybriden, *sequentiellen* Verfahren werden unterschiedliche Empfehlungsansätze nacheinander (sequentiell) durchlaufen. Dabei bildet der Output eines Ansatzes jeweils den Input des nachfolgenden Verfahrens, quasi wie beim *Pipelining*; der letzte Empfehlungsansatz liefert den finalen Output. Abbildung 57 visualisiert den Ansatz.



**Abbildung 57: Hybrider, sequentieller Ansatz**

Quelle: (Jannach 2011, S. 129)

Die Umsetzung des hybriden, sequentiellen Ansatzes kann mit folgenden Strategien erfolgen (vgl. Burke 2007):

- *Cascade Strategy*: In Form eines stufenweisen Prozesses wird zunächst durch eine erste Empfehlungstechnik eine Vorauswahl geeigneter Kandidaten getroffen. Diese Vorauswahl wird anschließend durch Anwendung einer zweiten Empfehlungstechnik zu einem Ergebnis weiterverarbeitet.
- *Meta-level Strategy*: Es wird ein Modell durch eine Empfehlungstechnik generiert, das den Input für die nächste Technik bildet. Insbesondere in Kombinationen von inhaltsbasierten und kollaborativen Techniken ist diese Strategie von Vorteil, da das Modell die Benutzerinteressen repräsentiert. (vgl. wieder Burke 2002, S. 9).

### 6.3.4 Wissensbasiertes Empfehlen

Die Grundidee wissensbasierter Empfehlungsverfahren besteht in der Verwendung von explizitem Domänenwissen, abgebildet in einer Wissensbasis. Das in der Wissensbasis abgebildete Wissen wird genutzt, um über Schlussfolgerungen die Bedürfnisse der Benutzer



zu ermitteln und entsprechende Empfehlungen zu generieren. Ramezani et al. unterscheiden drei Typen wissensbasierter Empfehlungssysteme (vgl. Ramezani et al. 2008, S. 1):

- *Case-based Reasoning-Systeme* nutzen als Basis vorhandene Informationen (Lösungen) früherer Situationen und erzeugen daraus Empfehlungen, indem die aktuelle Situation mit vergangenen Situationen verglichen wird.
- *Constraint-based Reasoning-Systeme* erzeugen Empfehlungen unter Berücksichtigung konkreter Bedingungen. Solche Bedingungen können beispielsweise konkrete Interessen oder Bedürfnisse eines Benutzers sein.
- *Rule-Based Systems* erzeugen Empfehlungen auf Basis definierter Regeln. Diese Regeln verbinden explizit die Interessen der Benutzer mit passenden Elementen.

Renneberg ergänzt die drei vorgestellten Typen durch einen weiteren Typ, dem Typ auf der Basis des sog. *ontologisch formulierten Wissens* (vgl. Renneberg 2010, S. 58); solche Systeme nutzen Ontologien als Basis für Empfehlungen. In den Ontologien wird explizites Bereichswissen deklarativ in Form von Entitäten und Beziehungen formuliert. Die Ontologien werden anschließend zur Erzeugung der Empfehlungen genutzt.

### 6.3.5 Demographisches Empfehlen

Bei demographischen Empfehlungen handelt es sich nicht um ein konkretes Empfehlungsverfahren oder eine klar definierte Methode; vielmehr werden Benutzerprofile durch demographische Informationen ergänzt, um ähnliche Benutzer zu ermitteln und daraus wiederum Empfehlungen zu erzeugen. Beispielsweise können Daten wie Geschlecht, Alter, Wohnort, Ausbildung, Familienstand und Einkommen zur Erzeugung von Profilen genutzt werden (vgl. Pazzani 1999, S. 395).

### 6.3.6 Kontextbasiertes Empfehlen

Kontextbasierte Verfahren nutzen Informationen über den situativen Kontext eines Benutzers zur Ermittlung von relevanten Informationsobjekten. Im Rahmen von Empfehlungssystemen können folgende Bereiche differenziert werden, in denen Kontextinformationen Anwendung finden (Adomavicius und Tuzhilin 2011 und Palmisano et al. 2008):

- Data Mining,
- E-commerce Personalization,
- Ubiquitous/mobile context-aware systems,

- Databases,
- Information Retrieval und
- Marketing and Management.

Im Umfeld des *Data Mining* besteht der Kontext beispielsweise aus Ereignissen, die Lebensabschnitte von Kunden charakterisieren. Diese Ereignisse können wiederum Auswirkungen auf die Eigenschaften der Kunden, ihren Status oder ihre Bedeutung für ein Unternehmen haben. Kenntnisse über derartige Informationen ermöglichen z. B. Mining Patterns eine Fokussierung auf relevante Daten bzw. das Filtern relevanter Ergebnisse<sup>67</sup>. E-Commerce-Plattformen verwenden beispielsweise Daten über Einkäufe als wichtige Kontextinformationen, weil über sie auf das Verhalten der Kunden geschlossen werden kann. Auf dieser Basis können weitere Produkte oder Dienste ermittelt werden, die möglicherweise für den Kunden auch von Interesse sein können (vgl. Palmisano et al. 2008).

Im Bereich ubiquitärer (allgegenwärtiger) und mobiler Systeme werden Kontextinformationen insbesondere im Zusammenhang mit ortsbezogenen Diensten eingesetzt. Dazu zählen beispielsweise Daten über Aufenthaltsort, Benutzer-Identität oder Temperatur. Auf Basis solcher Kontextinformationen können dem Benutzer für ihn relevante Orte in der Nähe ermittelt und empfohlen werden.

Auch Datenbanksysteme können von Kontextinformationen profitieren. Beispielsweise können die Systeme Abfragen unter Betrachtung des jeweiligen Kontextes, in dem die Abfrage gestellt wurde, unterschiedlich bearbeiten.

Innerhalb von Information-Retrieval-Systemen können Kontextinformationen helfen, Suchabfragen zu verbessern. Des Weiteren besteht die Möglichkeit, die Wahl des jeweiligen IR-Systems (wie z. B. Suchmaschinen oder Dokumenten-Management-Systeme) abhängig vom jeweiligen Kontext des Benutzers auszuwählen. Auch Entscheidungen im Rahmen des Marketings oder des Managements hängen häufig vom entsprechenden Kontext ab, innerhalb dessen sie getroffen werden.

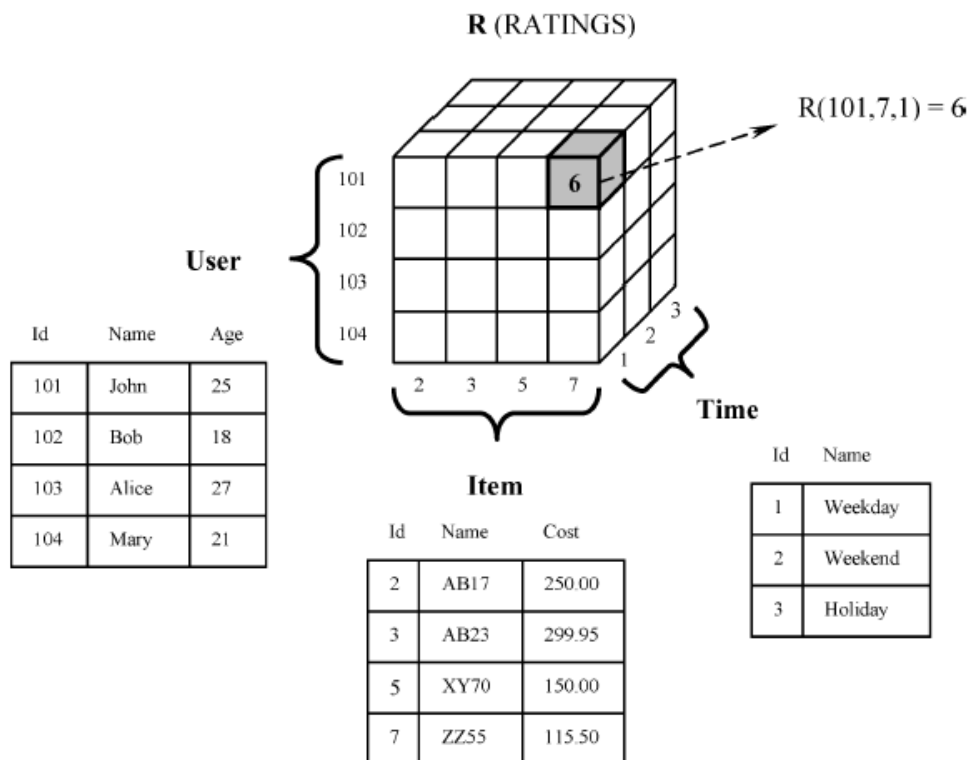
### 6.3.7 Multidimensionale Empfehlungssysteme

Wie in den vorherigen Abschnitten erläutert, unterstützen die klassischen Empfehlungssysteme die beiden Dimensionen Benutzer und Element. Multidimensionale Empfehlungssysteme sind in der Lage, weitere Dimensionen zu berücksichtigen. Beispielsweise stellen Adomavicius et al. mit dem MD-Modell (*Multidimensional Recommendation Model*) einen Ansatz vor, welcher analog zu Data Warehousing- bzw. OLAP-Konzepten

---

<sup>67</sup> Vgl. dazu auch (Linoff und Berry 2011).

mehrere Dimensionen einbezieht (vgl. Adomavicius et al. 2005). Beispielsweise kann die Zeit eine sinnvolle Dimension zur Ergänzung der Betrachtung von Benutzern und Elementen darstellen (siehe auch Abbildung 58).



**Abbildung 58: Multidimensionales Modell für einen „Benutzer-Element-Zeit“-Empfehlungsraum**

Quelle: (Adomavicius et al. 2005, S. 115)

Die Berücksichtigung mehrerer Dimensionen ermöglicht auch die Erzeugung unterschiedlicher Typen von Empfehlungen. Während die traditionellen Empfehlungssysteme lediglich Empfehlungen einer Form unterstützen (z. B. „Empfehlung der besten  $N$  Elemente für einen Benutzer“), kann durch die Ergänzung neuer Dimensionen die Empfehlung auf die Dimensionen fokussiert werden (z. B. „Empfehlung der besten  $N$  Benutzer/Zeit-Kombinationen für ein Element“ oder „Die Besten  $N$  Zeiten für eine Benutzer-/Element-Kombination“).

## 6.4 Bewertungen als Kernkonzept von Empfehlungssystemen

Bewertungen (häufig auch als *Ratings* oder *Feedback* bezeichnet) spielen im Rahmen von Empfehlungssystemen eine wesentliche Rolle, da sie (wie die vorherigen Abschnitte gezeigt haben) für viele Algorithmen notwendige Parameter repräsentieren. Unterschieden werden *explizite* und *implizite* Bewertungen (vgl. Hu et al. 2008, S. 1): Explizite Bewertungen sind konkretes Feedback der Benutzer zu einem bestimmten Element. Insbesondere in der Benutzer-Element-Matrix spiegeln sich explizite Bewertungen wieder. Schafer et al. unterscheiden drei Typen von Bewertungen in kollaborativen Systemen (vgl. Schafer et al. 2007, S. 293):

- *Scalar-Ratings*: Hierbei handelt es sich entweder um numerische oder um ordinale Werte. Bekannte Formen für numerische Bewertungen sind beispielsweise die „Sterne-Bewertung“, bei der einem Element eine Zahl (z. B. zwischen 0 und 5) zugewiesen wird. Ein Beispiel für ordinale Bewertungen ist die Skala „sehr zufrieden - eher zufrieden - eher unzufrieden - sehr unzufrieden“.
- *Binary ratings*: Es werden genau zwei Werte zugelassen. So kann ein Benutzer beispielsweise mit den Paaren „zufrieden/unzufrieden“ oder „gefällt/gefällt nicht“ bewerten.
- *Unary Ratings*: Es kann nur mit einem Wert bewertet werden, beispielsweise mit „gefällt mir“.

Explizite Bewertungen erfordern somit aktive Tätigkeiten der Benutzer. Implizite Bewertungen werden dagegen indirekt aus dem Benutzerverhalten gewonnen (vgl. Oard und Kim 1998, S. 1). Beispielsweise können vorherige Einkäufe, Surfverhalten oder gespielte Musiktitel Aufschluss auf die Präferenzen eines Benutzers liefern. Die Benutzer selber haben dabei keinen Aufwand, allerdings auch keinen aktiven Einfluss auf die Bewertungen.

Somit ergeben sich für beide Arten der Bewertungen Vor- und Nachteile, weswegen häufig eine Kombination beider Verfahren eingesetzt wird<sup>68</sup>. Bei expliziten Bewertungen können Benutzer beispielsweise bewusst oder unbewusst täuschen, in dem sie falsche Bewertungen abgeben. Implizite Bewertungen können durch unterschiedliche Umstände negativ beeinflusst werden, z. B. durch Einkäufe für Dritte. Da das System den Artikel dem

---

<sup>68</sup> Online-Shops wie z.B. Amazon nutzen eine Kombination expliziter und impliziter Verfahren: Für die direkte Bewertung der Produkte können „Sterne“ vergeben werden, was einer numerischen Bewertung (Scalar Ratings) entspricht. Über die vorherigen Einkäufe eines Benutzers werden ihm auch implizit Produkte empfohlen. Dabei kann die Basis der Empfehlung durch den Benutzer beeinflusst werden. So können Produkte (die beispielsweise für Dritte erworben wurden) aus der Liste entfernt werden.

Profil des einkaufenden Kunden zuordnet, wird für diesen Kunden ein Interesse an dem Produkt impliziert (siehe Abschnitt 6.5.4).

## 6.5 Einflussfaktoren auf Empfehlungen

Die unterschiedlichen Ausprägungen, Einsatzgebiete und Anwendungsszenarios von Empfehlungssystemen haben zur Folge, dass eine Reihe von Faktoren auf die Systeme wirkt, die den Empfehlungsprozess einerseits positiv, andererseits negativ beeinflussen können. Diese Faktoren werden im Folgenden näher betrachtet.

### 6.5.1 Datenbasis des Empfehlungssystems

Die Qualität der Empfehlungen und der damit zusammenhängende Nutzen eines Empfehlungssystems können in unterschiedlicher Art und Weise abhängig von dem zugrundeliegenden Datenbestand sein. Ist beispielsweise die Benutzer-Element-Matrix innerhalb eines kollaborativen Empfehlungssystems zu dünn besetzt, können keine repräsentativen Empfehlungen berechnet werden.

Existiert dagegen eine hohe Anzahl von Benutzer-Element-Bewertungen, besteht die Gefahr, dass eine Empfehlung aufgrund der Komplexität der Rechenoperationen nicht in Echtzeit durchgeführt werden kann. In diesem Zusammenhang wird auch vom sog. *Abdeckungsmaß* gesprochen, welches die Gesamtheit der vorhandenen Elemente bezeichnet bzw. die Gesamtheit der potentiellen Kandidaten (vgl. Celma 2010, S. 37)<sup>69</sup>.

### 6.5.2 Das „Cold Start“-Problem

Das so genannte *Cold Start Problem* tritt auf, wenn neue Objekte in das Empfehlungssystem aufgenommen werden. Nguyen et al. identifizieren bezugnehmend auf verschiedene Vorarbeiten (vgl. Burke 2002; Middleton et al. 2004, Schein et al. 2002 und Good et al. 1999) drei Typen von Cold-Start-Problemen (vgl. Nguyen et al. 2007, S. 1):

- neue Systeme,
- neue Elemente und
- neue Benutzer.

---

<sup>69</sup> Um der Problematik entgegenzuwirken, existieren Verfahren, die die Dimension der Matrix reduzieren, beispielsweise die *Singular Value Decomposition* und die *Non-negative Matrix Factorisation* (vgl. Celma 2010, S. 37).

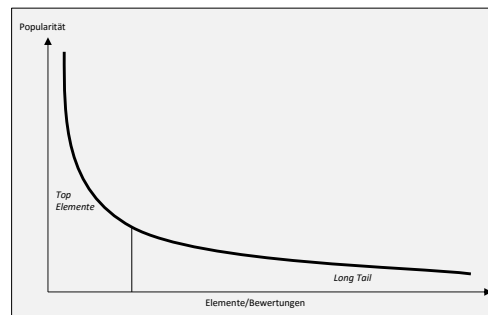
Das Cold-Start-Problem ergibt sich dadurch, dass das System keinerlei Kenntnisse über neu hinzugefügte Objekte besitzt. Neue Benutzer besitzen zunächst ein leeres Benutzerprofil und können erst Empfehlungen erhalten, wenn sie Aktionen durchgeführt haben. Bei kollaborativen Empfehlungssystemen können somit zunächst keine ähnlichen Benutzer ermittelt werden.

Ähnlich verhält es sich bei neuen Elementen. Auch hier können zunächst keine ähnlichen Elemente ermittelt werden, wenn das System nicht genügend Informationen über das jeweilige neue Element besitzt.

### 6.5.3 Das „Long Tail“- Problem

Der Begriff *The Long Tail* wurde 2004 von Chris Anderson geprägt, und beschreibt das Phänomen, dass Anbieter durch eine große Anzahl von Nischenprodukten einen hohen Gewinn erzielen können. Ermöglicht wird dies durch die Verbreitung des Internets, durch die geographische Beschränkungen weggefallen sind; die Beschränkungen waren bzw. sind auf „herkömmlichen“ Märkten dafür verantwortlich, dass Nischenprodukte häufig zu hohe Kosten verursachen (vgl. Anderson 2004). Anderson bezieht sich bei seiner Begriffsprägung auf eine exponentiell abfallende Häufigkeitsverteilung in der Statistik.

Bezogen auf Empfehlungssysteme besteht das Long-Tail-Problem darin, dass neue und unbekannte bzw. schwach bewertete Elemente häufig ignoriert werden (vgl. Park und Tuzhilin 2008), während existierende und beliebte Elemente in die Auswahlalgorithmen einbezogen werden. Dies führt dazu, dass die Empfehlungen auf einer kleinen Anzahl beliebter Elemente beruhen und dadurch die Beliebtheit der anderen Elemente weiter abnimmt. Abbildung 59 visualisiert den Zusammenhang: Die Y-Achse repräsentiert die Beliebtheit/Popularität, die X-Achse die Anzahl der Bewertungen. Es existieren wenige Elemente mit einer hohen Popularität und vielen Bewertungen, dagegen viele Elemente mit wenigen Bewertungen und geringer Popularität.



**Abbildung 59: The Long Tail**

Quelle: In Anlehnung an (Anderson 2004) und (Celma 2010)

Die Herausforderung für Empfehlungssysteme besteht folglich darin, auch diejenigen Elemente zu berücksichtigen, die sich innerhalb des sog. *Long Tails* befinden, da sich unter ihnen zu empfehlende Elemente befinden können (vgl. Park und Tuzhilin 2008).

### 6.5.4 Benutzerakzeptanz von Empfehlungen

Ein entscheidender Faktor dafür, inwieweit Empfehlungen von Benutzern akzeptiert werden, ist das Vertrauen in das entsprechende Empfehlungssystem und in die Art der Präsentation der Empfehlungen. Beispielsweise helfen Begründungen von Empfehlungen, die Akzeptanz zu steigern und somit die Nutzung der Empfehlungen des Systems zu steigern (vgl. Tintarev und Masthoff 2007).

Tintarev und Masthoff nennen sieben Möglichkeiten dafür, die Akzeptanz eines Empfehlungssystems und dadurch auch seiner Empfehlungen zu steigern (vgl. Tintarev und Masthoff 2007, S. 1):

- *Transparenz*: Erläuterung der Arbeitsweise des Empfehlungssystems.
- *Ergründbarkeit*<sup>70</sup>: Möglichkeit der Einflussnahme durch den Benutzer.
- *Vertrauen*: Steigerung des Vertrauens in das Empfehlungssystem.
- *Effektivität*: Unterstützung des Benutzers bei dessen Entscheidungen.
- *Überzeugungskraft*: Potenzial, Benutzer zur Nutzung (Kauf) oder zum Test zu überzeugen.
- *Effizienz*: Unterstützung des Benutzers, Entscheidungen schnell zu treffen.
- *Zufriedenheit*: Erhöhung der User Experience (der Freude an der Nutzung).

Transparenz kann erzielt werden, indem offengelegt wird, aus welchem Grund dem Benutzer ein bestimmtes Element empfohlen wird. Der Benutzer kann nachvollziehen,

<sup>70</sup> Übersetzt aus dem englischen „Scrutability“, was eine nichtexistierende Negation des Wortes „inscrutable“ (deutsch: unergründlich) darstellt.

weshalb ihm ein Element empfohlen wird, und dadurch besser bewerten, ob die Empfehlung für ihn relevant ist.

Ergründbarkeit bzw. Einflussname heißt, dass der Benutzer die Arbeitsweise des Empfehlungssystems beeinflussen kann, indem ihm die Möglichkeit geboten wird, bestimmte Parameter des Systems (z. B. Elemente oder Algorithmen) anzupassen.

Vertrauen in ein Empfehlungssystem kann sich somit einerseits aus der Transparenz der Arbeitsweise des Systems ergeben und andererseits aus der Möglichkeit zur („positiven“) Beeinflussung.

Häufig können Benutzer nicht direkt entscheiden, ob ein empfohlenes Element ihrem Bedarf entspricht. Daher kann es sinnvoll sein, sie davon zu überzeugen bzw. „dazu zu bringen“, ein empfohlenes Element auch zu „konsumieren“ (bei Produkten „zu kaufen“ bzw. „zu testen“). Umgekehrt kann eine hohe Anzahl ungeeigneter Elemente, die aufgrund einer solchen Beeinflussung konsumiert wurden, das Vertrauen in das Empfehlungssystem negativ beeinflussen (vgl. Tintarev und Masthoff 2007, S. 3).

Das zentrale Ziel von Empfehlungssystemen ist die Unterstützung des Benutzers bei seinen Entscheidungen. Die Unterstützung kann zum einen die Qualität der Entscheidungen betreffen (Effektivität), zum anderen die Geschwindigkeit, in der sie getroffen werden (Effizienz). Die Effektivität ist stark abhängig von der Genauigkeit der eingesetzten Empfehlungsalgorithmen.

Eine effektive Begründung einer Empfehlung unterstützt den Benutzer bei der Bewertung in Hinblick auf die Qualität und Relevanz der empfohlenen Elemente. Entscheidungen schneller zu treffen ist eines der meist verfolgten Ziele im Kontext von Empfehlungssystemen (vgl. Tintarev und Masthoff 2007, S. 3). Die Effizienz kann beispielsweise dadurch gesteigert werden, dass dem Benutzer die Unterschiede zwischen möglichen Alternativen dargestellt werden.

Auch das Nutzungserlebnis (*User Experience*) und die Benutzerfreundlichkeit (*Usability*) eines Empfehlungssystems beeinflussen die Akzeptanz des Benutzers. Beispielsweise wirkt sich eine übersichtliche und ausführliche Begründung einer Empfehlung positiv sowohl auf die Nützlichkeit als auch auf die Nutzbarkeit des Systems aus (vgl. Tintarev und Masthoff 2007, S. 3).

Studien zeigen, dass auch die grafische Präsentation einen erheblichen Einfluss auf die Glaubwürdigkeit eines Empfehlungssystems besitzt (vgl. Tintarev und Masthoff 2007, S. 2). Beispielsweise wurde herausgefunden, dass schon die Darstellung des Bildes eines



Autors bei einer Empfehlung die Glaubwürdigkeit und damit das Vertrauen bei den Benutzern erhöht.

### 6.5.5 Manipulation von Empfehlungen

Das Ergebnis von Empfehlungssystemen ist im Allgemeinen eine Liste von Elementen, die geeignet sind, einen Benutzer bei seiner Entscheidungsfindung zu unterstützen. Damit sind die Systeme ein attraktives Ziel für Angriffe, um sie dahingehend zu manipulieren, bestimmte Elemente zu empfehlen. Insbesondere kollaborative Empfehlungssysteme, bei denen Benutzer und Elemente als Eingabeparameter zur Berechnung der Empfehlungen genutzt werden, sind häufige Ziele von Angriffen und Manipulationen<sup>71</sup>.

Abhängig vom eingesetzten Empfehlungssystem und den in ihm implementierten Verfahren existieren unterschiedliche Möglichkeiten für einen Angriff.

Mobasher et al. unterscheiden vier Typen von Angriffen auf kollaborative Empfehlungssysteme (vgl. Mobasher et al. 2007, S. 5): Bei sog. *High-Knowledge-Attacks* besitzt der Angreifer detailliertes Wissen über die Arbeitsweise des Empfehlungssystems und richtet seine Manipulation explizit darauf aus, während bei *Low-Knowledge-Attacks* lediglich allgemeines Wissen über die Funktionsweise von Empfehlungssystemen besteht.

Als zweite Dimension betrachten Mobasher et al. die Absicht der Attacke. Mit *Push-Attacks* werden Angriffe bezeichnet, die die Beliebtheit eines Elementes erhöhen, bei *Nuke-Attacks* wird versucht, den Beliebtheitsgrad von bestimmten Elementen zu verringern.

Eine mögliche Form der Manipulation ist die sog. *Profile Injection*. Dabei werden konkrete Benutzerprofile in das System eingeschleust, um die Beliebtheit von Elementen zu erhöhen (Push) oder zu verringern (Pull). Ein Profil besteht aus einer Reihe von Bewertungen unterschiedlicher Elemente eines Benutzers (vgl. Benutzer-Element-Matrix, Abschnitt 6.3.1).

Im Hinblick auf eine Manipulation werden die Elemente folgendermaßen unterschieden (vgl. Mobasher et al. 2007, 5ff.):

- *Selected Items*: Explizit ausgewählte Elemente, die mit konkreten Werten bewertet werden.
- *Filler Items*: Mit zufälligen Werten bewertete Elemente.

---

<sup>71</sup> Zwischen einem „Angriff“ und einer „Manipulation“ wird im Rahmen dieser Arbeit nicht im Detail differenziert.

- *Unrated Items*: Nicht bewertete Elemente.
- *Target Item*: Einzelnes Ziel-Element, dessen Beliebtheitsgrad erhöht bzw. verringert werden soll.

Abhängig vom Ziel der Manipulation können die eingesetzten Profile unterschiedlich ausgeprägt sein. Für eine Übersicht unterschiedlicher Modelle und Typen sei auf Mobasher et al. verwiesen (vgl. Mobasher et al. 2007, 5ff.).

Auch bei anderen Empfehlungssystem-Varianten sind Manipulationen möglich. Beispielsweise können wissensbasierte Recommender durch bewusst fehlerhaft spezifizierte Regeln negativ beeinflusst werden. Durch Veränderung von Orts- oder Benutzer-Informationen können kontextbasierte Verfahren attackiert werden.

Um Angriffen und Manipulationen vorzubeugen bzw. sie zu verhindern, müssen adäquate Maßnahmen umgesetzt werden. Zusätzlich zu einer manuellen Prüfung können beispielsweise Algorithmen implementiert werden, die auffällige Verhaltens- oder Manipulationsmuster erkennen.

Zu beachten ist, dass die positive Auswirkung von Transparenz (siehe Abschnitt 6.5.4) einen negativen Einfluss auf die Manipulationsmöglichkeiten eines Empfehlungssystems besitzen kann. Denn die Kenntnis über die Arbeitsweise des Systems schafft zwar Vertrauen, ermöglicht jedoch auf der anderen Seite eine gezielte Angriffsstrategie (vgl. *High-Knowledge-Attacke*).

## 6.6 Zusammenfassung und kritische Betrachtung von Empfehlungssystemen

Die vorherigen Abschnitte haben einen Überblick über die Ziele und Eigenschaften der unterschiedlichen Empfehlungsverfahren gegeben sowie deren Einflussfaktoren betrachtet. Abbildung 60 zeigt zusammenfassend und in Form einer Empfehlungssystem-Taxonomie die unterschiedlichen Verfahren, Ebenen und Zusammenhänge. Detaillierte Informationen dazu haben Bobadilla et al. zusammengefasst (vgl. Bobadilla et al. 2013).

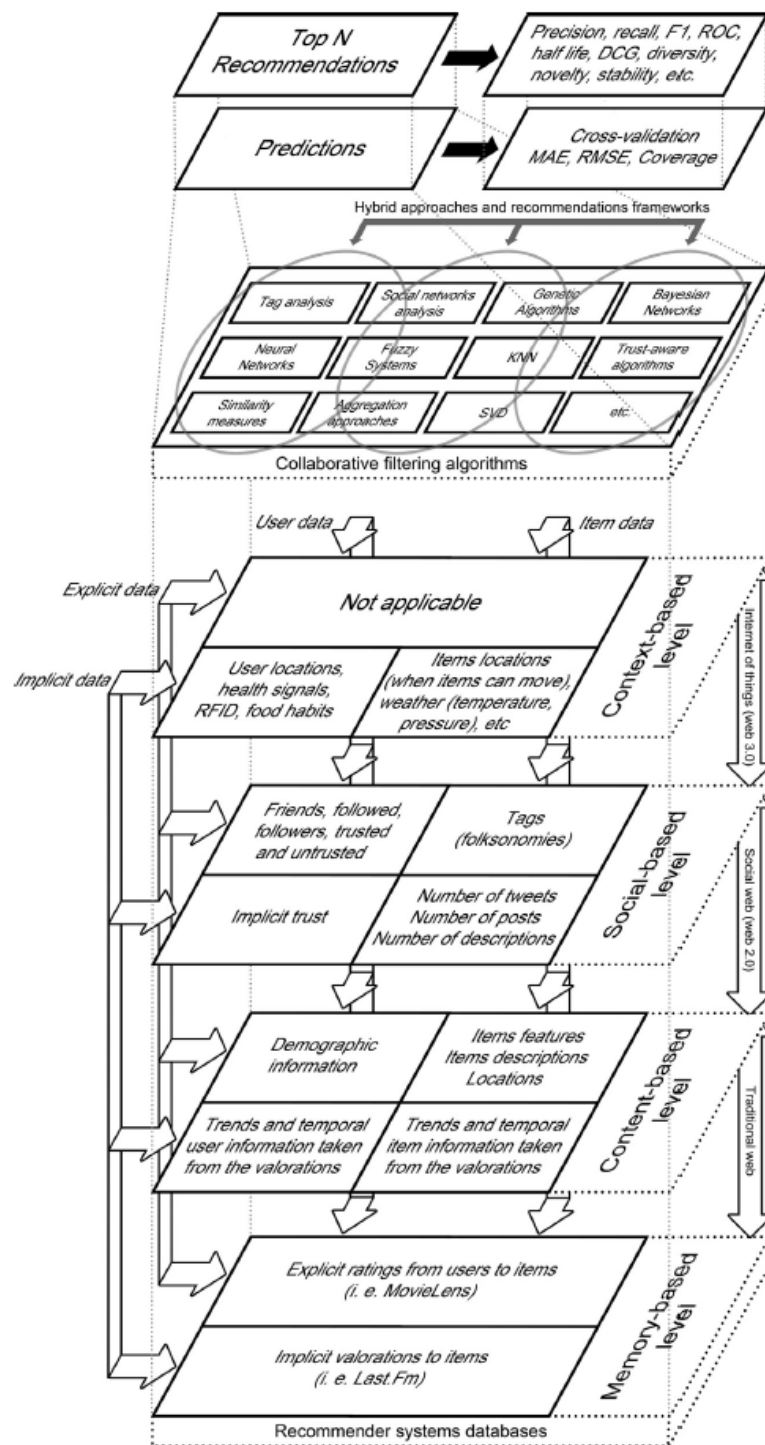


Abbildung 60: Empfehlungssystem-Taxonomie

Quelle: (Bobadilla et al. 2013, S. 126)

Kollaborative Empfehlungssysteme hängen von Benutzern und deren Empfehlungen ab. Für diese Art von Systemen ist es schwierig, neuen Benutzern geeignete Elemente zu empfehlen, da für sie noch kein Profil existiert (Cold-Start-Problem). Aber auch Benutzer mit einem untypischen Geschmack stellen ein Problem dar, da für sie nur wenige andere

Benutzer als *Nearest Neighbours* existieren; diese Problematik wird auch als *Gray Sheep* bezeichnet (vgl. Burke 2002, S. 4). Zudem besteht ein Nachteil des Long-Tails darin, dass in erster Linie Elemente empfohlen werden, die bereits beliebt sind, andere geeignete, aber bisher weniger bekannte Elemente, unberücksichtigt bleiben.

*Inhaltsbasierte Empfehlungssysteme* benötigen nicht den Input der Benutzer, um geeignete Elemente zu ermitteln, da bei ihnen die Ähnlichkeiten zwischen Elementen im Fokus der Berechnung liegen; dies führt zu einer „objektiven“ Betrachtung der Elemente. Jedoch bestehen auch bei diesen Systemen Startschwierigkeiten bei einer geringen Anzahl von Elementen und bei neuen Benutzern, die noch keine Elemente präferieren.

Sowohl inhaltsbasierte als auch kollaborative Verfahren besitzen den Vorteil, dass sie in der Regel automatisiert ablaufen, und Ähnlichkeitsbestimmungen von Elementen und Benutzern im Hintergrund durchgeführt werden können. Somit stehen die benötigten Informationen in Echtzeit zur Verfügung, sie sind jedoch anfällig gegenüber Attacken und Manipulationen.

Besonders aufwändig ist die Bestimmung relevanter Elemente bei Empfehlungssystemen, die zusätzliche Informationen und Dimensionen berücksichtigen (*multidimensionale bzw. kontextbasierte Empfehlungssysteme*). Diese Empfehlungssysteme sind jedoch in der Lage, qualitativ bessere Empfehlungen zu berechnen. Wenn sie nur Informationen verwenden, die durch ihre Benutzer nicht beeinflusst werden können, sind entsprechende Manipulationen ausgeschlossen.

Explizites Domänenwissen, definierte Regeln und Ontologien sind mögliche Basisdaten *wissensbasierter Systeme*. Der Aufwand zur Realisierung der Systeme ist dementsprechend hoch, gleichzeitig aber auch die Qualität der ermittelten Elemente. Für Manipulationen und Attacken bieten auch diese Systeme eine vergleichsweise kleine Angriffsfläche.

## 6.7 Empfehlungssystem innerhalb von PERCOP

Die zuvor beschriebenen Anwendungsfälle und Verfahren werden im Folgenden in Hinblick auf ihr Potenzial für die Nutzung innerhalb des intendierten Frameworks analysiert. Auf Basis der Analyse wird dann ein PERCOP-Empfehlungssystem entworfen; insbesondere werden geeignete Verfahren und Algorithmen eingebunden.

Die Bedeutung der erläuterten Empfehlungsansätze für PERCOP ist folgendermaßen zu bewerten: Bei *kollaborativen Verfahren* dient die Benutzer-Element-Matrix als Datenbasis. In PERCOP repräsentieren die Informationsartefakte die Elemente. Kollaborative Verfahren können in Situationen gute Ergebnisse liefern, in denen Benutzerrollen klar definiert sind: Es besteht eine hohe Wahrscheinlichkeit, dass eine Information, die für einen Benutzer relevant ist, auch für andere Benutzer mit der gleichen Rolle relevant ist.

Sind die Informationsartefakte durch Metadaten beschrieben oder durch andere Analysen semantisch erfasst, können *inhaltsbasierte Recommender* eingesetzt werden, die die Element-zu-Element-Korrelation<sup>72</sup> als Datenbasis nutzen. Solche Recommender können Ähnlichkeiten zwischen Elementen ermitteln und daraufhin gezielt Informationsartefakte empfehlen, die anderen (durch den Benutzer präferierte) Artefakte ähnlich sind und somit möglicherweise für den Benutzer auch relevant sein können.

Mit Hilfe des in PERCOP integrierten Regelsystems kann explizites Domänenwissen in Form von Regeln abgebildet werden. Diese Regeln ermöglicht die Erstellung und Nutzung von *wissensbasierten Empfehlungsverfahren*, welche als Datenbasis genau diese expliziten Empfehlungsregeln eines PERCOP-Frameworks nutzen können. Als Ergänzung zu Algorithmus-basierten Verfahren - welche Ähnlichkeiten zwischen Elementen oder Benutzern berechnen - können wissensbasierte Verfahren die Qualität der Empfehlungssysteme und dadurch auch der einzelnen Empfehlungen verbessern.

Auch für die Nutzung *kontextbasierter* und *multidimensionaler Empfehlungsverfahren* lassen sich im Rahmen von PERCOP sinnvolle Anwendungsfälle finden. Beispielsweise kann die Zeit als dritte Dimension in konkreten Domänen eine wichtige Rolle spielen. Dort könnte ein multidimensionaler Recommender die nötige Funktionalität bieten.

Bezugnehmend auf die zuvor beschriebenen Anwendungsfälle von Herlocker et al. bewertet Tabelle 12 die Bedeutung der Anwendungsfälle für PERCOP.

---

<sup>72</sup> Vgl. dazu auch (Schafer et al. 1999, S. 163).

Anwendungsfälle nach Herlocker et al.	Bewertung der Bedeutung für PERCOP	Relevanz
Annotation in Context	Das direkte Anreichern von Informationselementen durch Zusatzinformationen bietet hohes Potenzial für effektives Empfehlen.	+
Find Good Items	Als ein Hauptziel des Frameworks soll eine Liste geeigneter Informationen dem Benutzer zur Entscheidungsfindung bzw. Lösung einer Aufgabe präsentiert werden.	+
Find All Good Items	Das Auffinden aller Elemente zu einem gegebenen Input spielt im betrachteten Kontext eine untergeordnete Rolle. Durch den generischen Aspekt ist eine Umsetzung jedoch möglich.	-
Recommend Sequence	Gerade bei komplexen Aufgaben oder Problemstellungen kann die gezielte Präsentation von Informationen in einer bestimmten Reihenfolge einen sinnvollen Mehrwert bieten.	+
Just Browsing	Das "Stöbern" durch Informationsobjekte ohne konkretes Vorhaben ist im Rahmen des vorgestellten Frameworks nicht von Relevanz.	-
Find Credible Recommender	Das Vertrauen in Recommendern spielt eine große Rolle. Mechanismen zum Aufbau von Vertrauen werden benötigt (z. B. Transparenz, Reputationen etc.)	+
Improve Profile	Das System muss dem Benutzer vermitteln, dass sein Engagement ein eigener Vorteil bedeutet.	+
Express Self	Da innerhalb des Frameworks die Masse eine größere Rolle spielt als die Einzelperson, ist der Faktor Selbstdarstellung im Rahmen der Empfehlungen zunächst von geringer Bedeutung. Jedoch kann der Punkt bei der Erstellung von Informationsobjekten an Bedeutung gewinnen.	o
Help Others	Die Bereitschaft von Benutzern, anderen zu helfen ist wertvoll und muss bestmöglich gefördert bzw. verstärkt werden.	+
Influence Others	Die bewusste Manipulation durch falsche Bewertungen muss verhindert werden. Dies kann beispielsweise durch Reputationen, kollektive Intelligenz oder Algorithmen erfolgen.	+

+: Hohe Relevanz o: Mittlere Relevanz -: Geringe Relevanz

**Tabelle 12: Recommender-Anwendungsfälle in PERCOP**

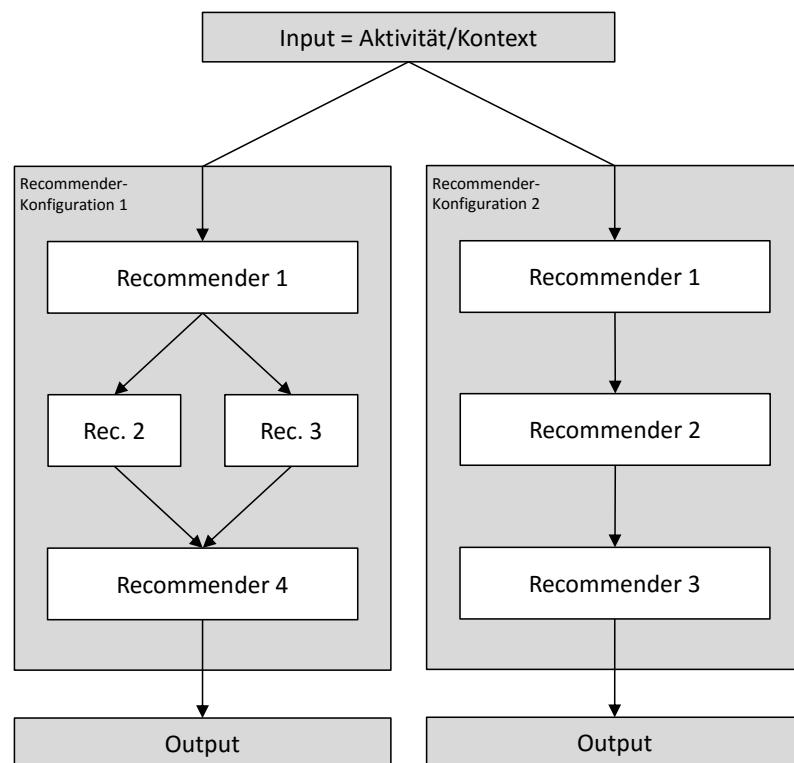
Ein PERCOP-Framework soll in unterschiedlichen Domänen eingesetzt werden. Deshalb muss ein solches Framework zum einen unterschiedliche Informationsquellen, Inputs und Empfehlungsverfahren unterstützen. Zum anderen muss das jeweils verwendete Empfehlungssystem in der Lage sein, in Abhängigkeit von der jeweiligen Domäne die passenden Verfahren bereitzustellen.

Daraus ergeben sich zusätzlich zu den bereits beschriebenen Anforderungen folgende zusätzliche Anforderungen:

- Unterstützung der Erstellung verschiedener Kombinationen aus Empfehlungsverfahren,
- Berücksichtigung unterschiedlicher Eingabedaten (Inputs),
- Auswahl und Durchführung in Echtzeit sowie
- eine fortlaufende Verbesserung der Verfahren.

Im Folgenden wird ein Empfehlungssystem präsentiert; es wurde so entworfen, dass alle ermittelten Anforderungen berücksichtigt werden. Insbesondere sind Empfehlungsverfahren frei konfigurierbar und die Ergebnisbewertung im Rahmen der Auswahl der konkreten Kombination wird berücksichtigt

Abbildung 61 zeigt den Aufbau des PERCOP-Empfehlungssystems; im Folgenden werden die Komponenten des Systems erläutert.



**Abbildung 61: Mehrstufiges Recommender-System**

Die Elemente *Recommender* und *Recommender-Konfiguration* repräsentieren die Kernkomponenten des (zweistufigen) Empfehlungssystems. Jeder Recommender muss in der Lage sein, für einen gegebenen Input ein Ergebnisset zu erzeugen. Bei der Recommender-Konfiguration handelt es sich um eine Komponente, welche verschiedene Recommender

beinhalten kann. Innerhalb der Konfiguration können die Recommender auf unterschiedliche Art und Weise kombiniert werden. Hier kommen die Konzepte der hybriden Empfehlungstechniken zum Einsatz.

Aus Nutzersicht unterscheiden sich Recommender und Recommender-Konfiguration kaum. Beide besitzen die gleichen Schnittstellen und ermitteln für unterschiedliche Eingabedaten geeignete Empfehlungen. Jedoch bietet diese Struktur folgende Vorteile:

- Recommender und Recommender-Konfigurationen können durch verschiedene Personen (Rollen) getrennt realisiert werden. Während bei Recommendern der technische Aspekt der Realisierung im Vordergrund steht, liegt bei der Recommender-Konfiguration der Fokus auf der fachlichen Kompetenz, die enthaltenen Recommender für die jeweilige Domäne/Umgebung bestmöglich zu kombinieren.
- Die Recommender-Konfiguration ermöglicht die Anpassung an die Eigenschaften und Anforderungen der zugehörigen Domäne bzw. Umgebung durch die bedarfsgerechte Kombination von geeigneten Recommendern.
- Die Recommender-Konfiguration kann als einzelnes Element verwaltet werden, was wiederum eine eigene Bewertung dieser ermöglicht.

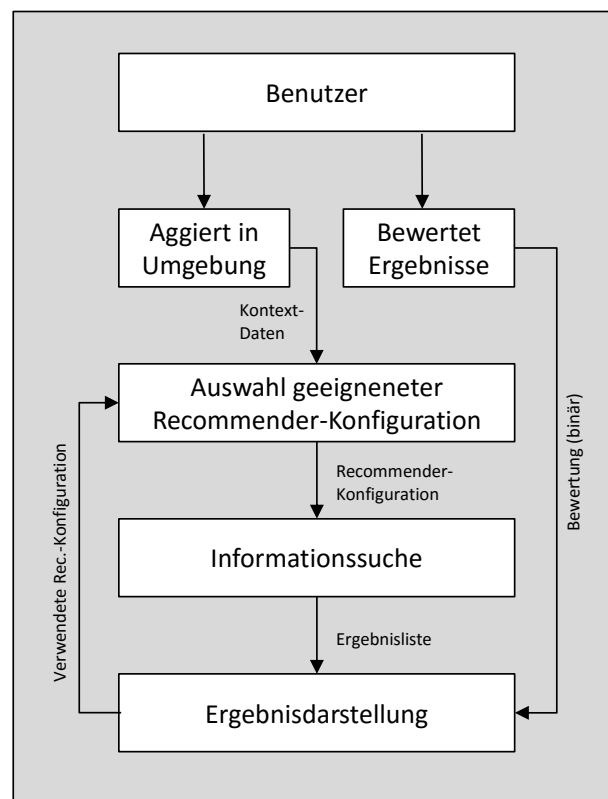
Ergebnis dieser Lösung ist ein „*Empfehlungssystem für Empfehlungssysteme*“, bei dem konkret der folgende Prozess durchlaufen wird:

Ein Benutzer führt eine Aktion (Funktionalität) aus, die als Input für die Suche nach geeigneten Informationen dient und die Suche anstößt. Abhängig von der durchgeführten Aktion wird eine konkrete Recommender-Konfiguration genutzt. Die aktuell eingesetzte Recommender-Konfiguration für jeden Benutzer wird gespeichert. Als Ergebnis erhält der Benutzer eine Liste von möglicherweise nützlichen Informationen.

Bewertet nun der Benutzer ein Element, wird nicht nur der jeweilige Kontext, Benutzer und das Element gespeichert, sondern auch die aktuell genutzte Recommender-Konfiguration. Somit können durch das System besonders geeignete Recommender-Konfigurationen ermittelt und vorgeschlagen bzw. automatisch verwendet werden, indem analysiert wird, bei welchen Recommender-Konfigurationen besonders viele Elemente als hilfreich bewertet wurden.

Das entwickelte System realisiert somit ein „*Empfehlungssystem für Empfehlungssysteme*“, welches implizit während der Nutzung durch die Bewertung der Elemente verbessert wird. Abbildung 62 visualisiert den Prozess des mehrstufigen PERCOP-Empfehlungssystems.





**Abbildung 62: Empfehlungsprozess innerhalb von PERCOP**

Bei der Erstellung von Recommendern und Recommender-Konfigurationen können die vorherigen Erläuterungen und die Analyse der Verfahren sowie die Gegenüberstellung (siehe Tabelle 12) zur Hilfe genommen werden. Abhängig von der jeweiligen Umgebung und den gegebenen Inputs (z. B. Datenquellen und Sensorwerten) können die geeigneten Verfahren implementiert und zu Recommender-Konfigurationen orchestriert werden.

Der präsentierte Ansatz integriert sich nahtlos in die PERCOP-Plattform, da das Empfehlungssystem in Teilkomponenten aufgeteilt wurde, die unabhängig voneinander durch unterschiedliche Mitglieder der Crowd realisiert werden können. Beispielsweise können Softwareentwickler Empfehlungskomponenten programmieren, die dann wiederum von anderen zu Empfehlungskonfigurationen für spezifische Domänen kombiniert werden können.

Im Folgenden werden zwei Beispiele für Recommender und Recommender-Konfigurationen präsentiert und deren Einsatz innerhalb des PERCOP-Frameworks beschrieben. Die Beispiele repräsentieren einen in der Praxis gängigen Bedarf an relevanten Informationen und sollen exemplarisch zeigen, dass das konzipierte Empfehlungssystem in der Lage ist, die Informationsbeschaffung zu unterstützen.

### *Szenario 1 (Softwareentwicklung)*

Im Bereich der Softwareentwicklung werden häufig Informationen bzgl. auftretender Fehler, eingesetzter Komponenten oder genutzter Technologien benötigt. Nützliche Quellen für Informationen sind Expertensysteme oder Foren wie beispielsweise *Stack Overflow*<sup>73</sup>. Eine sinnvolle Recommender-Konfiguration für diesen Anwendungsfall könnte folgendermaßen konstruiert werden:

- Zunächst wertet ein wissensbasierter Recommender auf der PERCOP-Plattform definierte Regeln aus.
- Parallel dazu existiert ein *StackOverflow-Recommender*, der im Expertensystem *Stack Overflow* nach passenden Beiträgen sucht.
- Das Ergebnis des *StackOverflow-Recommenders* wird als Input in einen kollaborativen Recommender weitergeleitet. Dort wird das Ergebnis in Hinblick auf ähnliche Benutzer (vgl. Konzept „Nächste Nachbarn“) evtl. noch einmal umsortiert.
- Anschließend werden die beiden Ergebnis-Sets (Wissensbasierter Recommender und kollaborativer Recommender) zu einem Ergebnis zusammengeführt, wobei die Ergebnisse aus den Regeln eine höhere Gewichtung erhalten.

In dem Szenario wird ein hybrider Ansatz spezifiziert, der sowohl die Strategie *pipelined* als auch *sequenziell* realisiert. Dieses Szenario wird innerhalb der Fallbeispiele noch einmal aufgegriffen (siehe Kapitel 9).

### *Szenario 2 (Interne Wissensdatenbank)*

In Unternehmen ist die Identifizierung und Beschaffung zugehöriger Dokumente zu einem konkreten Auftrag bzw. Kunden häufig ein aufwändiger Prozess. PERCOP und sein Recommender-Ansatz können diesen Prozess beispielsweise in folgender Weise unterstützen:

- Das Öffnen eines Dokuments durch einen Benutzer wird von einem Sensor erkannt und die Suche wird initiiert.
- Eine interne Wissensdatenbank oder eine Dokumentenmanagement-Software wird nach relevanten Dokumenten durchsucht.
- Parallel werden Regeln ausgewertet, und die Auswertung ergibt beispielsweise, dass bestimmte Informationen zu diesem Kontext nur dem aktuell agierenden Benutzer angezeigt werden.

---

<sup>73</sup> Siehe <http://stackoverflow.com>

- Anschließend werden beide Untersuchungsergebnisse an eine Recommender-Komponente weitergeleitet, die die Elemente filtert (z. B. nach relevanten Benutzern) und sortiert (z. B. zeitlich).
- Schließlich werden die Ergebnisse dem Benutzer präsentiert.

In diesem Szenario greift das PERCOP-Framework auf eine bestehende Lösung (Dokumentenmanagementsystem) durch die Nutzung spezieller Regeln und Empfehlungsverfahren zu. Dabei handelt es sich um einen mehrstufigen hybriden Ansatz. Zunächst wird die Variante *parallized* realisiert. Dazu werden zwei wissensbasierte Recommender kombiniert. Anschließend wird das Ergebnis einer multidimensionalen Recommender-Komponente als Input übergeben (*sequentieller, hybrider* Ansatz).

Die beiden Szenarios zeigen, dass die gewählte Art der Recommender-Integration eine sehr hohe Flexibilität ermöglicht, somit die generische Eigenschaft des Frameworks adäquat umsetzt. Insbesondere ermöglicht diese Form der Umsetzung weitreichende Optimierungen. Beispielsweise könnte der spezifizierte Stack-Overflow-Recommender als Ergebnisse nur Antworten des Forums vorschlagen, die von Mitgliedern als sog. *Accepted Solution* (Akzeptierte Lösung) markiert wurden. Somit würden nicht relevante Foren-Beiträge im Vorfeld durch den Recommender ausgefiltert, was zu einer schnelleren Problemlösung beiträgt.

Im Detail hängen die Optimierungen von der Art der Informationsquelle ab; dies ist Gegenstand des folgenden Kapitels.

## 7 Informationsquellen und -erzeugung

Gegenstand des siebten Kapitels ist die Analyse von Informationen in Hinblick auf die Relevanz und Nutzung im Kontext des PERCOP-Frameworks. Zunächst werden in Abschnitt 7.1 die unterschiedlichen Quellen von Informationen analysiert und die entsprechenden Informationsartefakte spezifiziert. Anschließend wird in Abschnitt 7.2 zunächst allgemein auf die Bedeutung der Kontextmodellierung und des Metadatenmanagements eingegangen. Anschließend werden konkrete Konzepte (Architektur und Prozess) zur Annotation von Informationsartefakten durch Metadaten für die Verwendung innerhalb von PERCOP vorgestellt. Abschnitt 7.3 schließt das Kapitel mit einer Zusammenfassung ab.

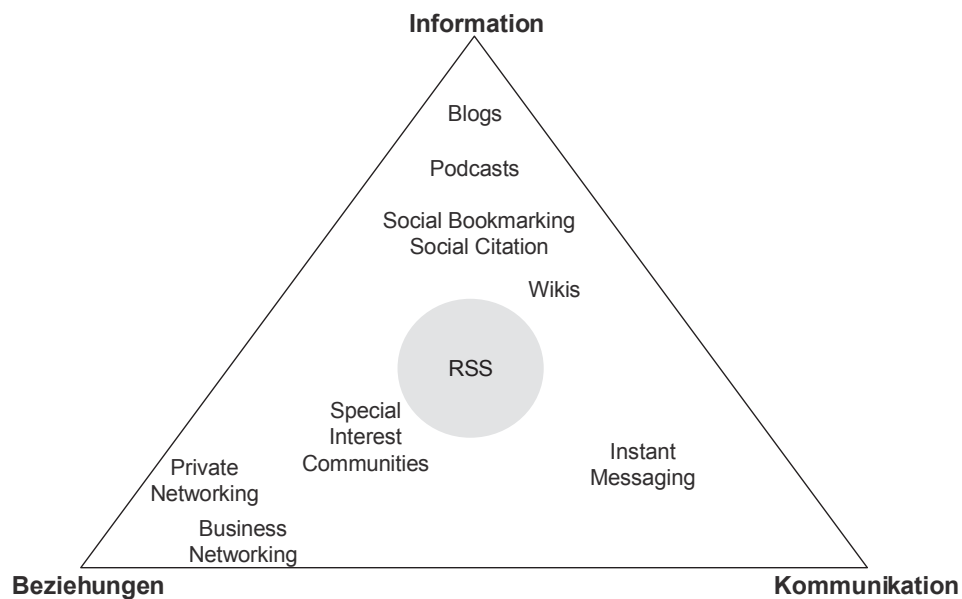
### 7.1 Informationsquellen

Ein konkreter Informationsbedarf ist abhängig von der jeweiligen Situation, in der sich ein Benutzer befindet. Im Kontext aufgabenzentrierter Informationssuche unterscheiden Byström und Järvelin beispielsweise zwischen faktenorientierten (*fact-oriented*), problemorientierten (*problem-oriented*) und allgemeinen (*general-purpose*) Informationsquellen (vgl. Byström und Järvelin 1995, S. 203).

Je nach Einsatz von PERCOP können unterschiedliche Informationsquellen von Relevanz für die Benutzer sein; allerdings sind elektronische und (online) verfügbare Quellen erforderlich, um eine Informationsbeschaffung und -bereitstellung in Echtzeit realisieren zu können.

„Kandidaten“ für entsprechende Quellen sind natürlich vor allem die im Rahmen des Web 2.0 entstandenen *Social Software* Systeme; dazu zählen *Blogs*, *Wikis*, *Podcasts*, *Special Interest Communities* und *Instant Messaging*.

Hippner ordnet die Systeme in ein dreidimensionales Schema ein (vgl. Hippner 2006, S. 9); die relevanten Dimensionen sind *Information*, *Beziehung* und *Kommunikation* (vgl. Abbildung 63). Die Elemente der Kategorien Beziehung und Kommunikation können beispielsweise genutzt werden, um Verbindungen zwischen Benutzern (Autoren) herzustellen oder Informationen aus Instant Messaging-Kommunikationen zu durchsuchen. Für PERCOP sind jedoch in erster Linie die Systeme mit Schwerpunkt in der Dimension Information relevant; Auf diese Elemente wird deshalb im Folgenden im Detail eingegangen.



**Abbildung 63: Klassifikationsschema von Social Software**

Quelle: (Hippner 2006, S. 9)

Koch und Richter beschreiben in Anlehnung an McAfee (vgl. McAfee 2006) folgende Charakteristika von Social Software, die auch im Rahmen der Informationsquellen von PERCOP zutreffen (vgl. Koch und Richter 2007, S. 14):

- So einfach wie möglich selbst Beiträge veröffentlichen oder Inhalte editieren können.
- Durch Tagging einfach strukturierende Metadaten beitragen können.
- Durch Annotations- und Verlinkungsmöglichkeiten einfach zusätzliche Inhalte und Metadaten bereitstellen können.
- Durch Abonnierungsmöglichkeiten einfach auf neue Inhalte aufmerksam gemacht werden können.
- Beigetragene Inhalte einfach auffindbar machen.
- Den Aufbau der Anwendungen modularer, dienstorientierter und datenzentrierter gestalten.

### 7.1.1 Weblogs<sup>74</sup>

Bei Weblogs (Kombination aus den englischen Wörtern „Web“ und „log“) handelt es sich um Webseiten, auf denen in regelmäßigen Abständen neue Inhalte veröffentlicht werden. Die gängige Kurzform für die Bezeichnung „Weblog“ ist *Blog*. Eine wesentliche Eigenschaft von Blogs ist, dass die Einträge umgekehrt chronologisch sortiert dargestellt werden (vgl. Griesbaum 2007, S. 3). Des Weiteren besitzen Blogs in der Regel folgende Eigenschaften (vgl. Holtz und Demopoulos 2006, 12ff.):

- Starke Verlinkung zu anderen Inhalten (Hyperlinks).
- Simulierte bidirektionale Hyperlinks (Trackback) ermöglichen rückwertige Verlinkung zur Verfolgung von Diskussionen.
- Darstellungsformen wie bspw. Feeds ermöglichen das Abonnieren von Blogs und die Integration in sog. FeedReader.
- Eine eindeutige Adresse jedes Beitrags (Permalink) unterstützt einfaches Weiterempfehlen und Kommentieren.
- Kommentare anderer Benutzer erscheinen unter jedem Beitrag und ermöglichen die Beteiligung anderer Benutzer.
- Empfehlungen und Verweise auf andere Blogs können durch den Betreiber durch sog. Blogrolls vorgenommen werden.
- Such- und Archiv-Funktionen ermöglichen das Auffinden älterer Beiträge.
- Ordnen von Beiträgen durch Tags und Kategorien.

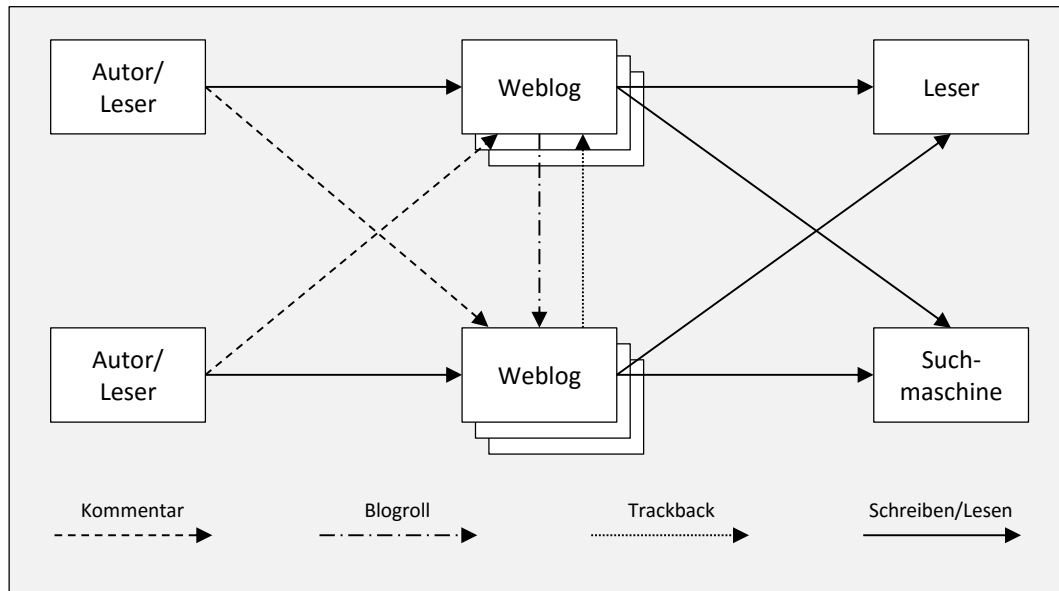
Blogs erfreuen sich einer hohen Beliebtheit, insbesondere deshalb, weil ihre Nutzung keine Kenntnisse in den Bereichen Web-Entwicklung und Web-Design voraussetzt. Zur Erstellung und Verwaltung von Blogs existiert eine Vielzahl von Web-Anwendungen, die eine einfache Bedienung und Anpassung ermöglichen.

Im Laufe der Zeit haben sich unterschiedliche Arten von Blogs entwickelt; insbesondere haben neben den persönlichen Blogs unternehmensinterne Blogs an Bedeutung gewonnen; je nach Zweck und Zielgruppe werden innerhalb dieser sog. *Corporate Blogs* Ausprägungen wie *Knowledge Blogs*, *Service Blogs*, *CEO Blogs*, *Product Blogs* und *Crisis Blogs* unterschieden (vgl. Zerfaß und Boelter 2005). Unterstützt wird durch diese Blogs ein gezielter Austausch von Informationen zu spezifischen Themengebieten wie z. B. Produkten, Dienstleistungen und Technologien.

---

<sup>74</sup> Die Ausführungen innerhalb dieses Abschnittes rekurrieren (wenn keine weiteren Quellen angegeben) auf Griesbaum (vgl. Griesbaum 2007).

Alle Blogs in Summe werden als *Blogosphäre* bezeichnet; Abbildung 64 zeigt eine Übersicht über die Bestandteile dieser Blogosphäre und ihre Beziehungen untereinander.



**Abbildung 64: Blogosphäre**

Quelle: (Alpar et al. 2007, S. 17)

Wegen ihrer starken Verbreitung und Nutzung, und aufgrund ihrer Eigenschaften sind Blogs für die Informationssuche im Kontext des Frameworks als hochrelevant einzustufen. Zu den Eigenschaften von Blogs, die für PERCOP wesentlich sind, zählen:

- Blogs sind nicht auf bestimmte Themen oder Anwendungsfelder festgelegt.
- Über Blogs sind Informationen durch Mitglieder der Crowd auf einfache Art und Weise zu erstellen und zu veröffentlichen.
- In Abhängigkeit von der zu betrachteten Domäne können Blog-Einträgen wertvolle Informationen bis hin zu Lösungswegen zu konkreten Problemstellungen enthalten.

Zudem erlauben Blogs die konkrete Suche nach Blog-Einträgen durch Suchmaschinen. Beispielsweise bietet Google einen Blogsuchdienst mit erweiterten Möglichkeiten wie die erhöhte Genauigkeit bei Datumsbereichen oder die Volltext- und die Tag-Suche. Diesbezüglich ergibt sich unmittelbar ein Nutzungsvorteil für PERCOP, da - wie angesprochen - Untersuchungen des Suchverhaltens von Anwendern gezeigt haben, dass die erweiterten Funktionen der Suchmaschinen kaum genutzt werden.

PERCOP ermöglicht es dagegen, in Abhängigkeit des Kontextes automatisch Blogs in die Suche mit einbeziehen. Dazu sind spezielle Recommender-Module zu integrieren, die auf

Blogs und ihre Eigenschaften zugeschnitten sind. In Kapitel 9 soll dieser Aspekt im Rahmen von Fallbeispielen detaillierter analysiert werden.

### 7.1.2 Fragen/Antwort-Seiten (Q&A-Websites)

Für den Austausch konkreter Lösungen zu Problemen bzw. für die Beantwortung von Fragen haben sich innerhalb der Social Software besondere Community-Webseiten entwickelt, sog. *Q&A-Websites*<sup>75</sup>. Bei Q&A-Websites handelt es sich um Community-Webseiten<sup>76</sup> bzw. -Foren, über die Benutzer Fragen stellen können. Andere Nutzer haben die Möglichkeit, auf diese Fragen zu antworten. Bekannte Q&A-Websites sind Quora<sup>77</sup>, Super User<sup>78</sup>, Yahoo! Answers<sup>79</sup>, Askville<sup>80</sup>, Answers.com<sup>81</sup> und Stack Overflow<sup>82</sup>.

Beispielsweise stellt Stack Overflow eine Q&A-Plattform für den Bereich der Software-Entwicklung getrennt nach den unterschiedlichen Technologien bereit. Mitglieder können nicht nur Fragen stellen und Fragen anderer Mitglieder beantworten, sondern können auch Antworten positiv oder negativ bewerten. Darüber hinaus können Antworten durch den Verfasser der Frage als „akzeptierte Lösung“ (engl. *accepted solution*) markiert werden; eine solche Markierung ermöglicht ein schnelles Auffinden von hilfreichen Antworten in der häufig großen Menge der Antworten anderer Mitglieder.

Die Mitglieder von Stack Overflow können durch ihre Beteiligung in Form von hilfreichen Antworten oder Fragen sog. *Reputation Points* erhalten. Neben dem Ansehen in der Community erhalten Benutzer abhängig von ihren Reputation Points besondere Privilegien bzw. erweiterte Funktionalität auf der Stack Overflow-Plattform. Auf diesen Aspekt wurde bereits oben im Rahmen der Betrachtung der Motivation im Detail eingegangen.

Stack Overflow stellt auch eine Programmierschnittstelle<sup>83</sup> bereit, mit der ein direkter Zugriff auf Plattformfunktionalität ermöglicht wird. Dadurch kann insbesondere auf folgende Elemente zugegriffen werden:

---

<sup>75</sup> Q&A steht im Englischen für *Question & Answer* (Frage und Antwort)

<sup>76</sup> Die Q&A-Webseiten können in Bezug zu Abbildung 63 in die Rubrik *Special Interests Communities* eingeordnet werden, da diese häufig bestimmte Domänen adressieren.

<sup>77</sup> Siehe <https://www.quora.com>

<sup>78</sup> Siehe <http://superuser.com>

<sup>79</sup> Siehe <http://answers.yahoo.com>

<sup>80</sup> Siehe <http://askville.amazon.com>

<sup>81</sup> Siehe <http://www.answers.com/>

<sup>82</sup> Siehe <http://www.stackoverflow.com>

<sup>83</sup> Siehe <https://api.stackexchange.com/docs>



- *Fragen*: Alle Fragen einer Seite; konkrete Fragen; Fragen, die in Beziehung stehen; unbeantwortete Fragen etc.
- *Antworten*: Alle Antworten einer Seite; Antworten auf eine konkrete Frage; Kommentare einer Antwort etc.
- *Beiträge*: Beiträge, unabhängig ob Fragen oder Antwort
- *Kommentare*: Alle Kommentare; konkrete Kommentare eines Beitrags etc.
- *Benutzer*: Benutzerinformationen, d. h. alle Beiträge, Kommentare, Tags eines Benutzers, seine Inbox, die ungelesenen Elemente etc.
- *Suche*: Suche nach bestimmten und erweiterten Kriterien sowie die Suche nach ähnlichen Beiträgen.

Die Programmierschnittstelle ermöglicht es, gezielt nach konkreten Beiträgen zu suchen. Im Kontext des Frameworks stellen Q&A-Webseiten eine wichtige Informationsquelle für die Informationsermittlung dar. Die konkrete Anbindung soll - wie angesprochen - in Form einer Recommender-Komponente erfolgen. Sie ermöglicht die Anbindung einer effizienten Suche nach nützlichen Informationen, indem beispielsweise nur Antworten mit positiven Bewertungen angezeigt werden bzw. nur Antworten, die jeweils als *accepted solution* markiert wurden.

Im Rahmen der Prototypentwicklung (Kapitel 8) soll zur Verifikation des Ansatzes ein *StackOverflow-Recommender-Modul* entwickelt werden, das in Verbindung mit einem Sensor innerhalb einer Entwicklungsumgebung direkt nach relevanten Informationen (Antworten) sucht, beispielsweise, wenn ein Fehler auftritt.

### 7.1.3 Wikis

Bei einem Wiki handelt es sich um eine „*Webapplikation, die es den Besuchern nicht nur ermöglicht, Inhalte auf einer Website hinzuzufügen, sondern auch die Inhalte anderer Besucher zu editieren*“ (Hippner und Wilde 2005, S. 442). Somit ermöglichen Wikis die Erstellung und Bearbeitung von Online-Dokumenten in kollaborativer Art und Weise. Die Inhalte der Wiki-Seiten sind stark miteinander verlinkt (sog. *Wiki-Links*), und analog zu Content-Management-Systemen und Blogs benötigt ein Nutzer keine speziellen HTML-Kenntnisse zur Erstellung oder Bearbeitung von Beiträgen, da eine (einfache) Wiki-Syntax für die Strukturierung und Formatierung der Wiki-Seiten sorgt.

Im Gegensatz zu Weblogs liegt der Schwerpunkt bei Wikis weniger auf der Kommunikation und Konversation aktueller Informationen, sondern vielmehr auf der Dokumentation von Wissen. Wikis bieten sich dann an, wenn „*eine eng gekoppelte oder wenig geplante Zusammenarbeit an Dokumenten erforderlich ist*“ (Koch und Richter 2007, S. 38).

Des Weiteren identifizieren Koch und Richter folgende Nutzungsmöglichkeiten für Wikis (vgl. Koch und Richter 2007, S. 38):

- Dokumentation von Wissen,
- Projektmanagement,
- Sammlung/Austausch von Informationen,
- Plattform für internationalen Expertenaustausch,
- Kulturelle Austauschplattform für Expatriaten und
- Personal Information Management (PIM).

Entsprechend sind Wikis in das Wissensmanagement einzuordnen, während Blogs eher als Kommunikationsmedium dienen.

### 7.1.4 Social Bookmarking

Als weitere *Social Software* haben Bookmarking-Dienste an Bedeutung gewonnen; bekannte Plattformen für Social Bookmarking sind beispielsweise Del.icio.us<sup>84</sup> und Mister Wong<sup>85</sup>. Bookmarking-Dienste fokussieren auf die „*Erfassung, Kategorisierung und Verwaltung eigener Web-Bookmarks*“ (Koch und Richter 2007, S. 47). Zudem sind die Vernetzung sowie die Zuordnung von Bookmarks zu Benutzern eine wichtige Eigenschaft von Bookmarks (vgl. ebenda).

Bei der Verlinkung werden die Bookmarks mit Schlagwörtern (*Tags*) versehen, um eine effiziente Verwaltung und damit ein schnelles Wiederfinden zu ermöglichen; auf diese Metadaten soll in Abschnitt 7.2 näher eingegangen werden.

Für PERCOP sind Bookmarks natürlich geeignete Informationsquellen bzw. ein geeigneter Link zu Informationsquellen. Zwar enthalten die Bookmarks selbst keine Informationen, in Kombination mit der Verschlagwortung und der Verbindung zu anderen Quellen und Benutzern können sie jedoch einen wertvollen Beitrag zur Suche nach relevanten Informationsartefakten leisten. Beispielsweise kann über die Empfehlungskomponente ein ähnlicher Benutzer ermittelt werden und im nächsten Schritt können seine Bookmarks analysiert und für die anstehenden Informationssuchprozesse genutzt werden.

---

<sup>84</sup> Siehe <http://www.delicious.com>

<sup>85</sup> Siehe <http://www.mister-wong.de>

### 7.1.5 Microblogging

Bei *Microblogging* handelt es sich um eine spezielle Art des Blogs, bei der die Nachrichtenlänge beschränkt ist. Der bekannteste Microblogging-Dienst ist Twitter<sup>86</sup>: Twitter ist ein Echtzeit-Informationsnetzwerk zur Verbreitung von Nachrichten, die Länge der einzelnen Nachrichten (sog. *Tweets*) ist auf 140 Zeichen begrenzt. Die Kommunikation wird nach dem Observer-Prinzip durchgeführt: Personen können die Nachrichten von Autoren abonnieren und erhalten diese dann automatisch; sie werden als *follower* des jeweiligen Autors bezeichnet.

Speziell für Unternehmen bieten Tweets die Möglichkeit, Informationen schnell an ihre Kunden zu übermitteln:

*„Twitter verbindet Unternehmen in Echtzeit mit seinen Kunden. Unternehmen nutzen Twitter, um aktuelle Informationen schnell mit den Menschen zu teilen, die sich für ihre Produkte und Dienstleistungen interessieren. Sie sammeln in Echtzeit Marktinformationen und Feedback, und bauen Beziehungen mit Kunden, Partnern und Entscheidungsträgern aus.“ (Twitter 2013)*

Je nach Situation können im Rahmen von PERCOP auch Microblogging-Informationen eine relevante Quelle bilden, beispielsweise in Szenarios, in denen die Aktualität der Informationen entscheidend ist. Die Anbindung ist in den meisten Fällen möglich; beispielsweise bietet Twitter eine Programmierschnittstelle<sup>87</sup> zur Einbindung in Applikationen. Solche Schnittstellen können durch Recommender-Module in PERCOP eingebunden und über Recommender-Konfigurationen mit anderen Recommender-Modulen kombiniert werden. Wird zum Beispiel im Rahmen einer Web-Suche ein Dokument eines konkreten Autors gefunden, könnten anschließend automatisch die Twitter-Nachrichten dieses Autors auf möglicherweise sinnvolle ergänzende Echtzeitinformationen durchsucht und solche Nachrichten präsentiert werden.

### 7.1.6 Weitere Informationsquellen

Über die bisher vorgestellten Informationsquellen hinaus existieren noch weitere Quellen; Beispiele sind Video-Plattformen (z. B. Youtube<sup>88</sup>, Vimeo<sup>89</sup>), Multimedia-Systeme, Dokumentenmanagementsysteme, Datenbanken und Dateisysteme.

---

<sup>86</sup> Siehe <http://www.twitter.com>

<sup>87</sup> Siehe <https://dev.twitter.com/docs/api>

<sup>88</sup> Siehe <http://www.youtube.com>

<sup>89</sup> Siehe <http://www.vimeo.com>

Durch den generischen Aufbau von PERCOP können grundsätzlich beliebige Informationsquellen auf einfache Art und Weise in das Ziel-Framework integriert werden. Dies gilt für existierende und für in der Zukunft entstehende Quellen.

## 7.2 Kontextmodellierung und Metadatenmanagement

Um bei der großen Anzahl der verfügbaren Informationen die relevantesten Artefakte identifizieren zu können, sollten die Informationsartefakte mit Metadaten angereichert werden (vgl. Adida et al. 2012). Bei Metadaten handelt es sich um *“Daten über Daten”* (vgl. Hansen und Neumann 2001, S. 1050); sie enthalten beschreibende Informationen über die Daten bzw. über die Informationsartefakte.

*“Metadaten ermöglichen die Beschreibung und Entdeckung von Informationsobjekten [...], ihre Nutzung [...] sowie Ihr Management.” (Krcmar 2010, S. 77)*

Unterschieden wird zwischen formalen und inhaltlichen Metadaten (vgl. Krcmar 2010, S. 78): Äußere Merkmale eines Informationsartefaktes werden durch formale Metadaten beschrieben; darunter fallen beispielsweise das Dateiformat, das Datum und der Autor des Artefakts. Inhaltliche Metadaten beziehen sich demgegenüber auf die Inhalte der Informationsartefakte; Beispiele sind ein Abstract oder Schlagwörter.

Je nach Technologie, Verfügbarkeit und Art der Information existieren unterschiedliche Ansätze für inhaltliche Metadaten. sie werden im Folgenden zunächst vorgestellt und dann auf ihre Nutzbarkeit für das PERCOP-Konzept analysiert.

### 7.2.1 Ontologien, Taxonomien und Folksonomien

*Ontologien* bezeichnen zum einen eine Menge möglicher Begriffe und deren Deutungen für eine semantische Annotation von Artefakten. Zum anderen beinhaltet eine Ontologie auch die inhaltliche Beziehung dieser Begriffe in Form von Inferenz- und Integritätsregeln (vgl. Krcmar 2010, S. 80). Studer et al. kombinieren die Definitionen von Gruber (vgl. Gruber 1993, S. 1) und von Borst (vgl. Borst 1997, S. 12) und definieren eine Ontologie als eine *„formal, explicit specification of a shared conceptualisation“* (Studer et al. 1998, S. 185).

Für Fensel sind die beiden folgenden Aspekte prägend für Ontologien (vgl. Fensel 2004, S. 4):

- Ontologien definieren formale Semantiken für Informationen, die eine Verarbeitung durch Maschinen ermöglichen.
- Ontologien definieren Semantiken aus der Realwelt, die (basierend auf einem gemeinsamen Vokabular) eine Verbindung von maschinenverarbeitbaren Inhalten mit der Bedeutung für den Menschen ermöglichen.

Als wesentliche Ziele von Ontologien werden Wissensaustausch (*Knowledge Sharing*), logische Folgerungen (*Logic Inference*) und die Wiederverwendung von Wissen (*Knowledge Reuse*) identifiziert (vgl. Wang et al. 2004, 1ff.).

Abhängig vom Spezialisierungsgrad können die folgenden Ontologien unterschieden werden (vgl. Guarino 1998, S. 9–10):

- Top-Level-Ontologie,
- Domänenontologie,
- Aufgabenontologie und
- Anwendungsontologie.

Eine Top-Level-Ontologie beschreibt sehr allgemeine Konzepte, die über alle Wissensdomänen gleich sind, wie beispielsweise Ort, Zeit, Ereignis oder Aktion. Top-Level-Ontologien werden auch als *Upper-*, *Generic-*, oder *Foundation-Ontologien* bezeichnet. Ziel dieser Ontologien ist die Bereitstellung einer semantischen Interoperabilität für die darunterliegenden Ontologien. Umgekehrt werden die Begriffe der Top-Level-Ontologie durch die darunterliegenden Ontologien spezialisiert.

Domänen- und Aufgabenontologien liegen unterhalb der Top-Level-Ontologie und beschreiben ein Vokabular für eine bestimmte Domäne (z. B. Automobil oder Medizin) bzw. für bestimmte Aufgaben (z. B. Verkauf oder Diagnose).

Anwendungsontologien spezialisieren sowohl Domänen- als auch Aufgabenontologien, und beziehen sich auf eine konkrete Applikation.

Verbreitete Sprachen für die Beschreibung von Ontologien sind RDF-Schema<sup>90</sup>, OWL<sup>91</sup> und Topic Maps<sup>92</sup>.

*Taxonomien* beschreiben (als Spezialform der Ontologien) ein hierarchisches Klassifikationsschema ohne logische Relationen (vgl. Alby 2008, S. 121). Im Rahmen des Managements von Informationen können Taxonomien verwendet werden, um die

---

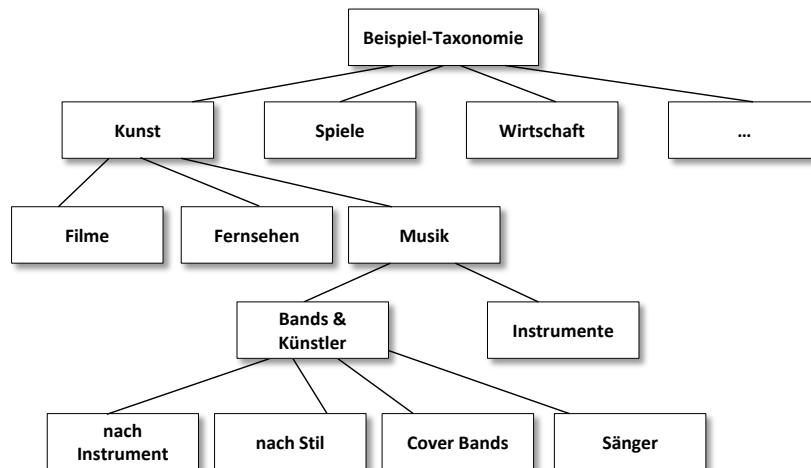
<sup>90</sup> Siehe <http://www.w3.org/TR/rdf-schema/>

<sup>91</sup> Siehe <http://www.w3.org/TR/owl-features/>

<sup>92</sup> Siehe <http://www.topicmaps.org>

unterschiedlichen Informationsartefakte in Kategorien einzuteilen, um ein Auffinden der Artefakte zu ermöglichen.

Abbildung 65 zeigt ein Beispiel für eine Taxonomie: Die Kategorie *Sänger* ist unterhalb der Kategorie *Bands & Künstler* angeordnet, die selbst eine Unterkategorie von *Musik* repräsentiert, die wiederum Teil der Kategorie *Kunst* ist.



**Abbildung 65: Beispiel einer Taxonomie**

Quelle: (Alby 2008, S. 122)

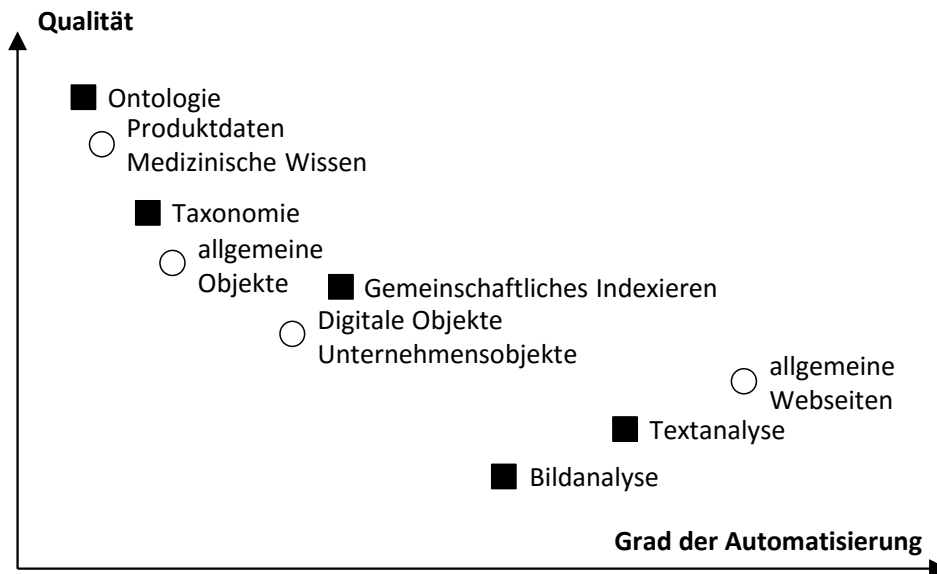
In eine solche Hierarchie können dann Informationsartefakte eingeordnet werden.

Die Eigenschaft, dass ihre Hierarchien festgelegt sind, bildet häufig einen Kritikpunkt für Taxonomien, da durch sie nicht alle Perspektiven und persönlichen Unterschiede berücksichtigt werden können (vgl. Alby 2008, S. 122).

Im Rahmen des Web 2.0 ist wegen diesem Kritikpunkt eine weitere Methode entwickelt worden, um Informationsartefakte zu klassifizieren, die sog. *Folksonomie* (vgl. Alby 2008, 121ff.). Bei einer Folksonomie handelt es sich um eine Sammlung von Schlagwörtern (Tags), die entsteht, wenn eine große Anzahl von Personen Informationsartefakte mit frei wählbaren Tags versehen. Diese Tags stehen im Gegensatz zur hierarchischen Taxonomie in keiner Beziehung zueinander und sind frei wählbar. Im Rahmen vieler Anwendungen im Bereich der Social Software (vgl. *Weblogs*, *Communities* und *Q&A Pages*) wird dieses Konzept des Tagging, das häufig auch als *gemeinschaftliches Indexieren* bezeichnet wird (vgl. Alby 2008, S. 127), eingesetzt.

Back et al. haben die Konzepte in Hinblick auf Qualität und Grad der Automatisierung graphisch angeordnet (vgl. Abbildung 66). Die Abbildung verdeutlicht, dass das Konzept

der Ontologien die höchste Qualität aufweist, ihm jedoch gleichzeitig der niedrigste Grad der Automatisierung zugemessen wird.



**Abbildung 66: Methoden zur Indexierung von Wissen**

Quelle: (Back et al. 2009, S. 43)

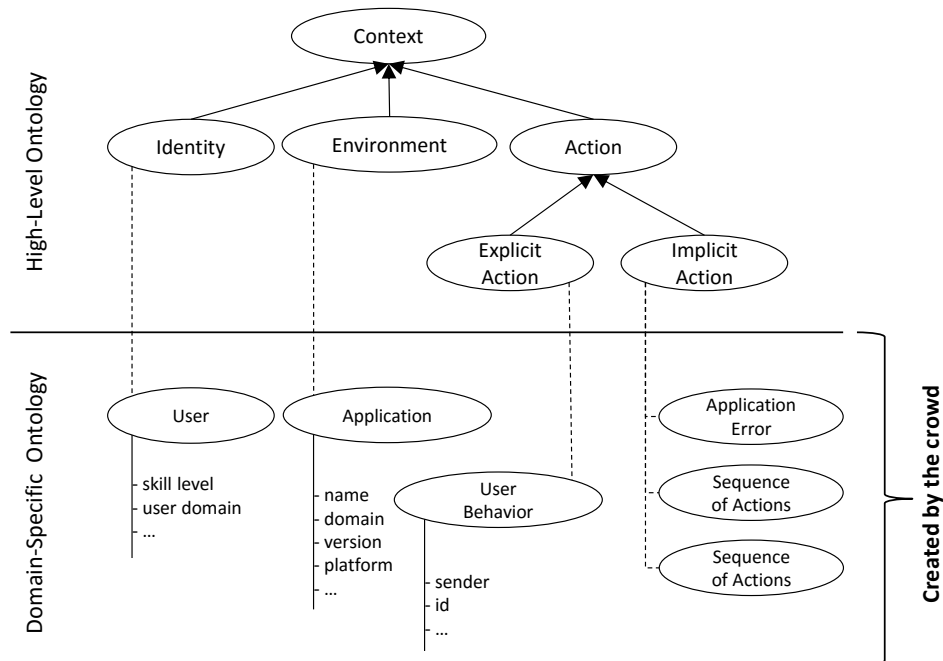
Im Rahmen des Einsatzes von PERCOP muss für den konkreten Fall entschieden werden, welches Konzept zur Modellierung von Kontextinformationen jeweils am sinnvollsten ist. Da die Ontologie das komplexeste der drei Verfahren bildet, wird im Folgenden auf dessen Nutzung näher eingegangen.

PERCOP unterstützt den Einsatz von Ontologien, um domänenunabhängige Varianten von Kontextdefinitionen definieren zu können. In Anlehnung an CONON<sup>93</sup> zeigt Abbildung 67 eine Ontologie zur Abbildung von Kontextinformationen bzgl. dem Benutzer und dessen Rolle (Identity), der Umgebung (Environment) und der ausgeführten Aktivitäten oder Funktionalitäten (Action).

Die High-Level-Ontologie definiert die allgemeinen Konzepte und kann durch domänenspezifische Ontologien spezialisiert werden: Die Umgebung (*Environment*) kann durch die Applikation selbst oder durch eine Kombination von Applikationen repräsentiert werden, und beinhaltet beispielsweise Informationen über *Name*, *Hersteller*, *Domäne* und *Version*. In einer konkreten Instanz des Frameworks wäre dann z. B. die Applikation

<sup>93</sup> CONON repräsentiert eine erweiterbare Kontext-Ontologie zur Modellierung von Kontext in *Pervasive-Computing-Umgebungen* (vgl. Wang et al. 2004, S. 18–22).

*Eclipse IDE* eine konkrete Umgebung mit spezifischen Eigenschaften. Das Objekt *Identity* könnte beispielsweise in einem konkreten Fall durch die Klasse *Benutzer* spezialisiert werden mit den Unterklassen *Status*, *Rolle* und *Skill-Level*.



**Abbildung 67: PERCOP-Ontologie**

Quelle: In Anlehnung an (Beul und Eicker 2009, S. 462)

Das *Action*-Element repräsentiert innerhalb von PERCOP eine Änderung des aktuellen Kontextes durch die Ausführung einer Aktivität oder durch die Nutzung einer Funktionalität durch den Benutzer. Die Ausführung kann entweder explizit oder implizit erfolgen (weshalb eine domänenspezifische Ontologie zwei Unterklassen „Explizit“ und „Implizit“ implementieren kann): Eine explizite Aktion wird aktiv durch den Benutzer durchgeführt, beispielsweise die Auswahl eines bestimmten Menüeintrags. Implizite Aktionen können durch die Umgebung angestoßen werden, z. B. durch Fehlermeldungen oder durch eine Zustandsänderung der Umgebung. Aber auch eine Sequenz oder Kombination von expliziten Aktionen kann als (implizite) Aktion definiert werden, beispielsweise die Durchführung von Benutzer-Aktionen in einer bestimmten Reihenfolge.



## 7.2.2 Semantische Annotationen auf Ressourcen-Ebene

Semantische Annotation (engl. *Semantic Annotation*) bezeichnet „den Prozess des Hinzufügens von semantischen Meta-Daten zu Dokumenten, die den Inhalt eines Dokuments in maschinen-verarbeitbarer Form beschreiben“ (Reif 2006, S. 1).

Scerri et al. unterscheiden interne und externe Annotationen (vgl. Scerri et al. 2005, S. 1): Interne Annotationen bezeichnen die Einbettung von semantischen Metadaten direkt innerhalb der (HTML-)Dokumente. Bei externen Annotationen werden die semantischen Deskriptoren in einer externen Quelle gespeichert, beispielsweise in einer Datei bzw. einer Datenbank. Das Dokument enthält lediglich einen Verweis auf die externe Quelle.

Im Folgenden werden einige ausgewählte Konzepte zur semantischen Annotation auf Ebene der Ressource vorgestellt. In Abschnitt 7.2.3 erfolgt die Darstellung der Adaption bzw. Nutzung innerhalb des Frameworks.

### 7.2.2.1 Mikroformate

Bei Mikroformaten (engl. *microformats*) handelt es sich um einen einfachen Ansatz für die semantische Annotation von Inhalten des WWW.

*“Microformats are a way of adding simple markup to human-readable data items such as events, contact details or locations, on web pages, so that the information in them can be extracted by software and indexed, searched for, saved, cross-referenced or combined.”  
(microformats.org 2013)*

Ziel ist es somit, Inhalte von Web-Dokumenten in einfachster Form zu kennzeichnen, zum einen, um die maschinelle Auswertung der Informationen zu ermöglichen, zum anderen, um ein schnelles Auffinden zu unterstützen. Zu solchen Inhalten gehören beispielsweise Kontakt-, Termin- oder Orts-Daten (vgl. Maurice 2009).

Die Grundidee von Mikroformaten besteht in der Nutzung von bestehenden Attributen der HTML-Dokumente bei der Anreicherung durch Metadaten; insbesondere die Attribute *class*, *rel* und *ref* werden für die Annotation verwendet. Beispielsweise ergänzt die folgende Formatierung einen Teil des Mikroformats *hcard* zu einem bestehenden HTML-Dokument:

```
<div id="hcard-MB" class="vcard">  
  <span class="fn">Michael Beul</span>  
  <span class="org">Universität Duisburg-Essen</span>  
</div>
```

Die Angaben *vcard* und *fn* innerhalb der Class-Attribute schaffen die Verknüpfung zu dem Mikroformat ohne wesentliche Veränderung des Dokuments, erlaubt aber eine einfache maschinelle Erkennung und Verarbeitung der semantischen Informationen.

### 7.2.2.2 RDFa

Mit Hilfe von RDFa (RDF in Attributes) können RDF-Modelle oder -Graphen in Web-Dokumente (HTML, XHTML, XML) eingebettet werden (vgl. Adida et al. 2012). Wie bei Mikroformaten werden auch bei RDFa die bestehenden Dokumente „lediglich“ um maschinenlesbare Informationen ergänzt; im Unterschied zu den Microformaten ermöglicht RDFa jedoch durch die Nutzung eines großen Spektrums an Sprachmitteln die Integration quasi beliebiger Metadaten.

Der Großteil der RDF-Statements lässt sich mit Hilfe der zusätzlichen Attribute *vocab*, *prefix*, *resource*, *property* und *typeof* modellieren. Das folgende Beispiel zeigt die entsprechende Ergänzung eines HTML-Dokuments um RDFa-Metadaten:

```
<div vocab="http://schema.org/" typeof="Person">
  Der Autor heißt <span property="givenName">Michael</span>
  <span property="familyName">Beul</span>
  
</div>
```

In dem Beispiel wird das Vokabular von [schema.org](http://schema.org)<sup>94</sup> verwendet. Die Attribute *vocab* und *typeof* legen fest, dass sich die nachfolgenden Angaben auf das Subjekt Person beziehen, dessen RDF-Bezeichner sich innerhalb des Namensraums von <http://schema.org> befinden.

### 7.2.2.3 Microdata und Schema.org

Eine weitere Möglichkeit, HTML-Dokumente durch die Integration von Annotationen semantisch zu beschreiben bildet Microdata<sup>95</sup>, eine HTML-Spezifikation der *Web Hypertext Application Technology Working Group* (WHATWG). Durch Erweiterungen des HTML-Standards unterstützt HTML5 die Elemente *itemscope*, *itemtype*, *itemid*, *itemref* und *itemprop*.

---

<sup>94</sup> Siehe <http://schema.org>

<sup>95</sup> Siehe <http://www.w3.org/TR/microdata>

Itemscope gibt an, dass eine Beschreibung verwendet wird. Mit Hilfe des `itemtype`-Elements kann angegeben werden, um welches Element es sich konkret handelt bzw. wie dieses Element zu beschreiben ist. Weit verbreitete Quellen (Adressen) für Beschreibungen im Internet sind [schema.org](http://schema.org)<sup>96</sup> und [data-vocabulary.org](http://data-vocabulary.org)<sup>97</sup>. Das `itemprop`-Element klassifiziert den markierten Text (bzw. URLs oder Zeitangaben). `Itemid` erlaubt die Zuweisung einer eindeutigen ID, mit dem `itemref`-Element können logische Verknüpfungen hergestellt werden. Das Beispiel aus dem vorherigen Abschnitt (RDFa) ist mit Microdata wie folgt zu realisieren:

```
<div itemscope itemtype="http://schema.org/Person">Der Autor heißt  
  <span itemprop="givenName">Michael</span>  
  <span itemprop="familyName">Beul</span>  
    
</div>
```

Microdata hat gegenüber Microformats den wesentlichen Vorteil, dass es durch die Trennung von Semantik und Design zu keinen Kollisionen kommen kann. So kann beispielsweise der Fall, dass Informationen innerhalb eines HTML-Dokuments weit auseinander stehen, bei Microformats zu Problemen führen, da es in diesem Fall schwierig ist, die entsprechenden Elemente korrekt mit einem HTML-Attribut zu umschließen. Durch das `itemref`-Element kann dieses Problem bei Microdata nicht auftreten.

### 7.2.3 Metadatenmanagement in PERCOP

Für das PERCOP-Framework ist die Integration von kontextrelevanten Informationen in Form von Metadaten direkt in die Quelldokumente zu fordern, da solche Metadaten das Auffinden relevanter Informationen erleichtern und zudem den Relevanzgrad erhöhen. Im Folgenden wird deshalb die Anreicherung webbasierter Daten um kontextspezifische Informationen für PERCOP analysiert. Im Rahmen der Forschungen für die vorliegende Arbeit wurde vom Autor dieser Dissertation eine Diplomarbeit (vgl. Czehak 2010) vergeben und betreut. Die Aufgabe bestand darin zu analysieren, wie konkret bezogen auf PERCOP webbasierte Daten durch Metadaten-Management um kontextspezifische Informationen angereichert werden können.

---

<sup>96</sup> Siehe <http://www.schema.org>

<sup>97</sup> Siehe <http://www.data-vocabulary.org>

### 7.2.3.1 Semantische Annotationen

In diesem Abschnitt soll die Möglichkeit der Integration von semantischen Annotationen in die relevanten Dokumente anhand von Beispielen erprobt werden. Dazu soll ein passendes Mikroformat-Profil entwickelt und eine RDFa-Integration auf Basis eines RDF-Annotationsschemas erstellt werden.

class	-	HTML4 definition of the 'class' attribute. This metadata profile defines some 'class' attribute values (class names) and their meanings as suggested by a draft of "Hypertext Links in HTML".
author	o	Author of the information; May be represented by nested hCard record. See section 4.8.4.2 of RFC 2445.
content	o	The content of the information.
dtstamp	o	Date/time of when the document containing information was created.
type	o	Type of the information. Note: this property may be repeated (information may belong in several types). Common values are: TUTORIAL, EXAMPLE etc.
application	o	name (The name of the application) version (The version of the application) domain (The domain of the application) manufacturer (The manufacturer of the application) action (The executed application action/functionality)
rel	-	HTML4 definition of the 'rel' attribute. Here is an additional value.
tag	o	Indicates that the referred resource serves as a "tag", or keyword/subject, for the enclosing PERCOPContext element.

**Abbildung 68: Beispiel für ein XMDP-Profil für Kontext-Informationen in PERCOP**

Quelle: In Anlehnung an (Beul und Eicker 2009, S. 461)

Abbildung 68 demonstriert, wie HTML-Dokumente mit Hilfe weniger Attribute zielgerichtet durch semantische Informationen angereichert werden können. Das entwickelte Profil umfasst Angaben zu Autor, Umgebung und Art des Dokuments. Abbildung 69 zeigt, wie menschenlesbarer Text mit Hilfe dieses Mikroformats auf einfache Art und Weise als maschinenlesbarer Inhalt spezifiziert werden kann; in der Abbildung werden die hervorgehobenen Werte den Attributen des Mikroformats entsprechend zugeordnet.

## Human readable

This tutorial was posted on April 18th, 2009 from Michael Beul, University of Duisburg-Essen addresses the domain(s) DesktopPublishing, GraphicalEditing, WebDesign and the application(s) Adobe Photoshop CS3/CS4.  
Keywords: filter, aquarelle, retouch, montage

```

type      dtstamp    author(org)  author(fn)  tags
  ↑         ↑         ↑         ↑         ↑

```

## Machine readable

This **tutorial** was posted on **April 18th, 2009** from **Michael Beul, University of Duisburg-Essen** addresses the domain(s) **DesktopPublishing, GraphicalEditing, WebDesign** and the application(s) **Adobe Photoshop CS3/CS4**.  
Keywords: **filter, aquarelle, retouch, montage**

Four arrows point downwards from the top of the diagram to the following labels:

- application domain
- application manufacturer
- application name
- application version

### Abbildung 69: Verwendung eines Kontext-Mikroformats

Quelle: (Beul und Eicker 2009, S. 462)

Die Nutzung eines Mikroformats unterstützt die Ziele des Frameworks insbesondere in Hinblick auf Einfachheit. Die Autoren der Dokumente sind in der Lage, schnell die nötigen Metadaten hinzuzufügen. Aber auch andere Mitglieder der Crowd können so bestehende Inhalte um semantische Daten ergänzen.

Wie oben beschrieben haben Mikroformate gegenüber anderen Konzepten wie beispielsweise RDFa Nachteile in Hinblick auf die Darstellungsmächtigkeit. Entsprechend dem generischen Anspruch von PERCOP müssen andere Konzepte in das Framework integrierbar sein. Abbildung 70 demonstriert eine solche Integration am Beispiel eines RDF-Ontologie-Modells, das mit Hilfe von RDFa in ein Web-Dokument eingebunden wird.

```

<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML+RdFa 1.0//EN" "http://www.w3.org/MarkUp/DTD/xhtmll-
rdfa-1.dtd">

<html xmlns="http://www.w3.org/1999/xhtml" version="XHTML+RdFa 1.0">

<head><title>S/W-Bildbearbeitung in Photoshop</title></head>

<body
  xmlns:dcterms=http://purl.org/dc/terms/
  xmlns:percop=http://www.personalcopilot.com/
  xmlns:percopenv=http://www.personalcopilot.com/environments/
  xmlns:photoshopcs6=http://www.personalcopilot.com/environments/photoshopcs6
  xmlns:rdf=http://www.w3.org/1999/02/22-rdf-syntax-ns#
  xmlns:xsd="http://www.w3.org/2001/XMLSchema" >

  <div about="http://www.pstipps.de/swretouch" typeof="percop:tutorial">

    <h1>
      <span property="dcterms:title">S/W-Bildbearbeitung in Photoshop</span>
      <span property="dcterms:description" content="Diese Seite bietet ..."/>
    </h1>

    <h2 property="dcterms:creator">Max Mustermann</h2>

    <h3>Datum: <span property="dcterms:date" content="2011-05-21"
      datatype="xsd:date">21.05.2011</span></h3>

    <h3>Dieses <span rel="percop:domain"
      resource="http://www.personalcopilot.com/#dtp"></span><span rel="percopenv:application"
      resource="http://www.personalcopilot.com/environments/#photoshopcs6"></span> Tutorial handelt
      von den Anwendungsmöglichkeiten der <span rel="photoshopcs6:action"
      resource="http://www.personalcopilot.com/environments/photoshopcs6/#swretouch">Schwarz/Weiß
      Bildbearbeitung </span>in <span property="percopapps:appmanufacturer">Adobe</span><span><span
      property="percopapps:appname">Photoshop</span> in der Version <span
      property="percopapps:appversion">CS6</span><span property="percopapps:appdomain"
      content="dtp"/>.

      Das Tutorial richtet sich an <span rel="photoshopcs6:skilllevel"
      resource="www.personalcopilot.com/environments/photoshopcs6/#beginner">Anfänger</span> der
      Anwendung Photoshop und ist besonders interessant für Mitarbeiter in der <span
      rel="photoshopcs6:userdomain"
      resource="http://www.personalcopilot.com/environments/photoshopcs6/#pictureediting">
      Bildbearbeitung</span>.</h3>

    <p> Hier folgt das Tutorial... </p>

  </div></body></html>

```

**Abbildung 70: RDFa-Integration in PERCOP**

In Anlehnung an (Czehak 2010, S. 57)

Das Beispiel basiert auf einem RDF-Annotationsschema, welches die Menge an Prädikaten festlegt, „die zur Annotation von Subjekt-Ressourcen eines bestimmten Typs für einen bestimmten Anwendungszweck verwendet werden sollen“ (Czehak 2010, S. 57). Tabelle 13 zeigt das in Abbildung 70 verwendete Annotationsschema.

Annotationsschema		
Subjekt ( <a href="http://pstipps.de/">http://pstipps.de/</a> )	Prädikat	Objekt
<b>Allgemeine Angaben</b> @prefix dcterms: <a href="http://purl.org/dc/terms">http://purl.org/dc/terms</a>	creator title description date	„Max Mustermann“ „S/W-Bildbearb. ...“ „Diese Seite...“ „2011-05-21“
<b>Anwendungsbereich</b> @prefix percop: <a href="http://www.personalcopilot.com/">http://www.personalcopilot.com/</a>	domain	dtp
<b>Angaben über die Umgebung</b> @prefix percopenv: <a href="http://www.personalcopilot.com/environments/">http://www.personalcopilot.com/environments/</a>	environment	„#photoshopcs6“
<b>Optionale Angaben über die Umgebung</b> @prefix percopapps: <a href="http://www.personalcopilot.com/environments/">http://www.personalcopilot.com/environments/</a>	appname appversion appdomain appmanufacturer	„Photoshop“ „CS6“ „DTP“ „Adobe“
<b>Anwendungsspezifische Ontologie</b> @prefix photoshopcs6: <a href="http://www.personalcopilot.com/apps/photoshopcs6/">http://www.personalcopilot.com/apps/photoshopcs6/</a>	action/ functionality skilllevel userdomain	„#swretouch“ „#beginner“ „pictureediting“
<b>Typisierung der Ressource</b> @prefix rdf: < <a href="http://www.w3.org/1999/02/22-rdf-syntax-ns#">http://www.w3.org/1999/02/22-rdf-syntax-ns#</a> >	type	percop:tutorial

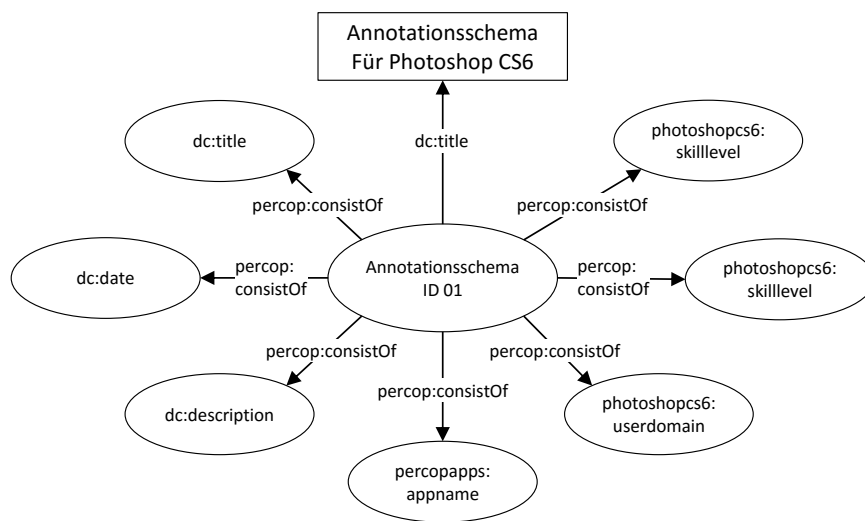
**Tabelle 13: Beispiel eines PERCOP-Annotationsschemas (RDF)**

Quelle: In Anlehnung an (Czehak 2010, S. 57)

Die allgemeinen Metadaten (Autor, Titel etc.) werden in dem Demonstrationsbeispiel mit Hilfe des weit verbreiteten Vokabulars *Metadata Terms* der DCMI (Dublin Core Metadata Initiative)<sup>98</sup> beschrieben. Durch spezifische Attribute wie *skilllevel*, *domain* oder *functionality* wird der Bezug zu dem jeweiligen Kontext hergestellt. Abbildung 71 visualisiert das Annotationsschema in Form eines RDF-Graphs<sup>99</sup>.

<sup>98</sup> Siehe <http://dublincore.org/documents/dcmi-terms/>

<sup>99</sup> Aus Darstellungsgründen zeigt die Abbildung einen Ausschnitt des Schemas.



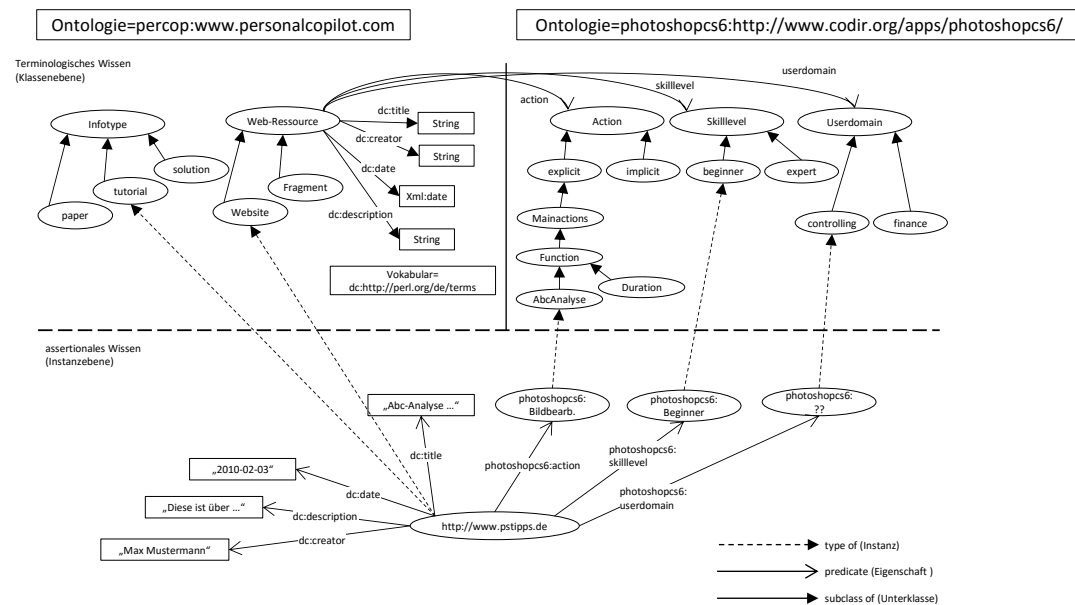
**Abbildung 71: PERCOP-RDF-Graph (Ausschnitt)**

Quelle: In Anlehnung an (Czehak 2010, S. 58)

Die Beispiele zeigen, dass RDF (RDFa und RDF-Schema) ein geeignetes Konzept für die Beschreibung von Web-Ressourcen mit kontextspezifischen Metadaten darstellt. Die Annotationsschemata können zentral verwaltet werden und erlauben eine direkte Integration in die Informationsartefakte.

Müssen im Rahmen der Wissensrepräsentation komplexe Ontologien abgebildet werden, eignet sich das RDF-Schema allerdings nicht; dann müssen darstellungsmächtigere Sprachen wie z. B. OWL eingesetzt werden. Dazu sei auf die Arbeit von Czehak (Czehak 2010) verwiesen, in der er detailliert auf die unterschiedlichen Konzepte eingeht. Abbildung 72 demonstriert auf der Basis von Czehak's Darstellungen an einem konkreten Beispiel die Möglichkeit, mehrere Ontologien miteinander zu verbinden.





**Abbildung 72: Verbindung von Ontologien in PERCOP<sup>100</sup>**

Quelle: In Anlehnung an (Czehak 2010, S. 74)

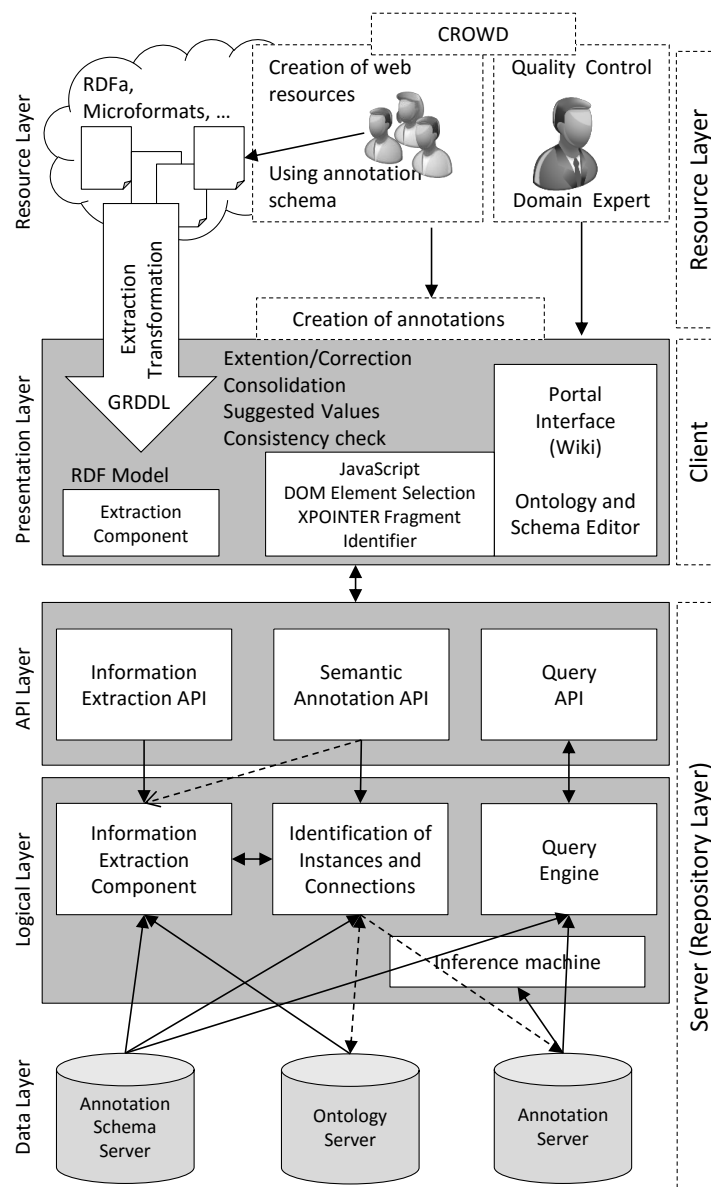
Ontologien für PERCOP können auf unterschiedliche Art und Weise realisiert werden, beispielsweise in Form von domänenspezifischen Ontologien, welche auf übergeordneten Ontologien (*Upper Ontologies*, vgl. Abschnitt 7.2.1) basieren. Die Ontologien sowie die Annotationen und -Schemata sollten auf speziellen Servern gesichert und publiziert werden, um den Zugriff von einem PERCOP-Framework entsprechend zuverlässig realisieren zu können.

Basierend auf den beschriebenen Verfahren zur Annotation von Web-Ressourcen und der Verwendung von Ontologien in PERCOP wird im folgenden Abschnitt der Prozess der Annotation spezifiziert. Dazu wird zunächst eine *Semantic Annotation Platform* vorgestellt und anschließend der zugehörige Prozess.

### 7.2.3.2 Annotationsarchitektur und -prozess

Abbildung 73 visualisiert eine Architektur zur Unterstützung der semantischen Annotation von webbasierten Dokumenten. In Anlehnung an Reeve und Han (vgl. Reeve und Han 2005) ergänzt Czehak Aspekte von PERCOP (vgl. Czehak 2010). Dabei soll die Architektur im Wesentlichen eine Orientierung über die mögliche Verwendung und Anordnung von Komponenten einer *Semantic Annotation Platform* liefern.

<sup>100</sup> Eine größere Darstellung befindet sich im Anhang der Arbeit



**Abbildung 73: Architektur einer *Semantic Annotation Plattform***

Quelle: In Anlehnung an (Czehak 2010, S. 116)

Die Architektur unterscheidet zum einen die Ressourcen- und die Repository-Ebene, zum anderen beinhaltet sie vier logische Schichten. Die *Präsentationsschicht* repräsentiert die Schnittstelle zu den Benutzern und stellt Funktionalitäten zur Annotation von Dokumenten bereit. Eine Möglichkeit ist die Realisierung über ein Browser-PlugIn, welches den Vorteil bietet, innerhalb derselben Umgebung sowohl die Dokumente zu betrachten als auch zu annotieren. Des Weiteren befindet sich eine Extraktionskomponente für eingebettete Metadaten als auch eine Schnittstelle, um Ontologien und Schemata zu verwalten.

Die Präsentationsschicht agiert direkt mit der *API-Schicht*. Diese Schicht kapselt die Funktionalitäten der *Logikschicht*. Insbesondere betrifft dies die Extraktions-, Annotations- und Abfrage-Funktionalitäten. Die zugehörigen Komponenten befinden sich auf der Logikschicht. Diese wiederum kommunizieren über Schnittstellen mit den unterschiedlichen Servern auf der *Datenschicht*.

Der konkrete Prozess auf Basis der beschriebenen Architektur läuft wie folgt ab (vgl. auch Abbildung 74 und Czehak (vgl. Czehak 2010, S. 119)):

- *Auswahl des Annotationsschemas*: Ein Benutzer betrachtet eine Web-Dokument und erkennt, dass dieses Dokument im Rahmen von PERCOP eine relevante Information für einen konkreten Anwendungsfall darstellt. Der Benutzer wählt dazu direkt innerhalb des Browsers das entsprechende Annotationsschema aus.
- *Laden des Annotationsschemas*: Das Schema wird automatisch aus der Annotationsschema-Datenbank in das Browser-Plugin geladen. Anschließend wird dynamisch auf Basis der zugrundeliegenden Ontologie-Konzepte und Schemata eine Eingabemaske für den Benutzer erzeugt.
- *Ermittlung von Vorschlagswerten*: Dem Benutzer werden Vorschläge für die Annotation generiert. Dabei können unterschiedliche Verfahren eingesetzt werden, beispielsweise die Analyse der eingebetteten Metadaten oder des Inhalts. Auch die Nutzung bereits vorhandener Schemata ist denkbar.
- *Eingabe der Werte*: Die Eingabe der Werte durch den Benutzer erfolgt durch manuelle Eingabe oder durch Übernahme der vorgeschlagenen Werte (siehe Punkt 3).
- *Bestätigung und Validierung*: Im Letzten Schritt wird der Annotationsprozess abgeschlossen. Es können noch Sichtbarkeitseinstellungen (privat/öffentlich) und Validierungen durchgeführt werden.

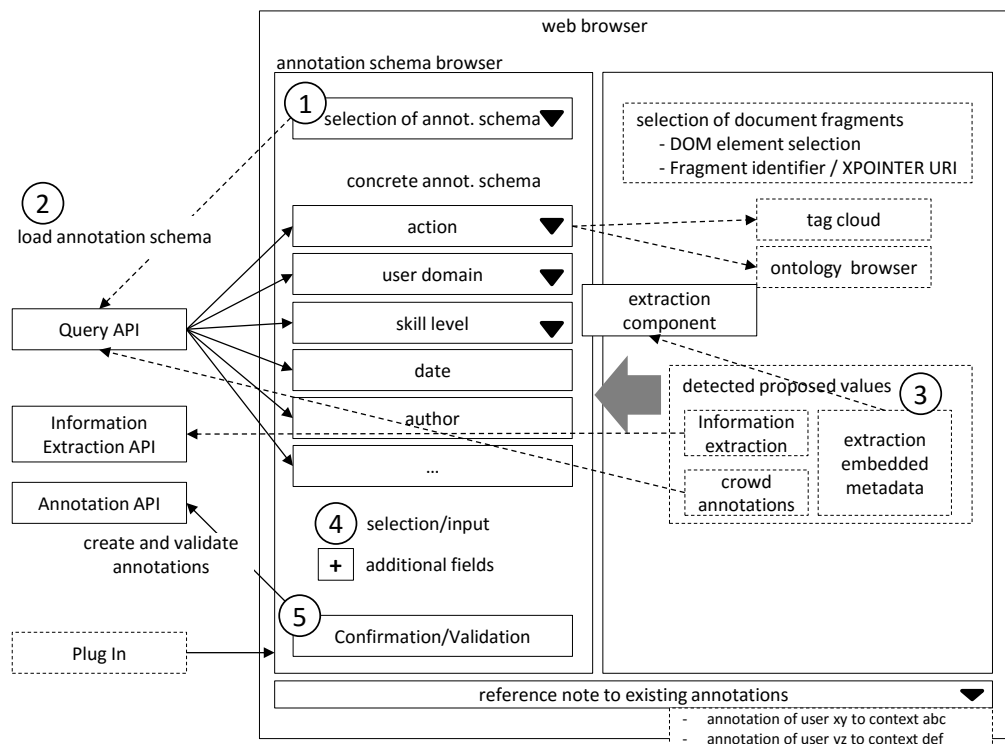


Abbildung 74: Annotationsprozess von Web-Dokumenten

Quelle: In Anlehnung an (Czehak 2010, S. 119)

## 7.3 Zusammenfassung

Das Kapitel hat mögliche Informationsquellen für PERCOP ermittelt sowie die Erzeugung von Informationsartefakten für das Framework analysiert. Da PERCOP die Bereitstellung von Echtzeitinformationen adressiert, stellen Internet-Dokumente die wesentliche Quelle für Informationen dar; zu solchen gehören insbesondere Webseiten, Wikis, Weblogs und Foren.

Um relevante Dokumente effektiv „finden“ zu können, sollten die Dokumente möglichst weitgehend durch Metadaten semantisch angereichert werden. Deshalb wurden die Möglichkeiten zur semantischen Annotation und ihre Integration in PERCOP analysiert. Es wurden eine Architektur sowie ein Prozess entworfen; der Prozess schließt die intendierte Beteiligung der Crowd bei der Annotation von Dokumenten ein, und erfüllt die Anforderung der Einfachheit.

# TEIL III

## Validierung des PERCOP-Frameworks

Der dritte Teil der Arbeit befasst sich mit der Validierung des erarbeiteten Frameworks. In Kapitel 8 wird zunächst ein Prototyp vorgestellt, der als konkrete Implementierung des Frameworks implementiert wurde. Anschließend wird der Prototyp bei ausgewählten Fallbeispielen eingesetzt (Kapitel 9).

## 8 Prototypische Umsetzung

Im Rahmen der für die vorliegende Dissertation durchgeführten Forschungsarbeiten wurde ein Prototyp als Proof of concept entwickelt. Durch die prototypische Umsetzung des entworfenen Frameworks soll die (technische) Umsetzbarkeit nachgewiesen werden. Des Weiteren soll über den konkreten Einsatz des Prototyps exemplarisch demonstriert werden, dass eine zielgerichtete Anwendung des Prototyps möglich ist (siehe Fallbeispiele in Kapitel 9); Basis bilden dazu die in Abschnitt 1.3 beschriebenen Szenarios.

Kapitel 8 ist wie folgt aufgebaut: Zunächst werden in Abschnitt 8.1 die wesentlichen Anforderungen an den zu entwickelten Prototyp definiert. Anschließend wird in Abschnitt 8.2 die Gesamtarchitektur entworfen, in den Abschnitten 8.3-8.8 werden dann die Komponenten der Architektur im Detail spezifiziert. Das Kapitel schließt mit einer Zusammenfassung (Abschnitt 8.9).

### 8.1 Anforderungen an den Prototyp

Ziel der Entwicklung des Prototyps ist die Abbildung der erarbeiteten Konzepte in eine konkrete Softwarelösung. Zum einen sollen alle wesentlichen Bestandteile des PERCOP-Frameworks umgesetzt werden, um die Realisierbarkeit nachzuweisen. Zum anderen bildet er die Basis für die durchzuführenden Fallbeispiele.

Entsprechend der allgemeinen Anforderungen an das Framework (Abschnitt 3.2), den spezifizierten IR-Strategien (Abschnitt 4.4), der intendierten Beteiligung der Crowd (Abschnitt 5.6.4) und den ausgewählten Empfehlungsverfahren (Abschnitt 6.7) sind folgende funktionale Anforderungen an den Prototyp zu stellen:

1. Es muss eine zentrale Plattform in Form eines Web-Portals bereitgestellt werden.
2. Innerhalb der Web-Plattform müssen Verwaltungsmöglichkeiten für die Hauptentitäten zur Verfügung gestellt werden (Umgebungen, Sensoren, Funktionalitäten, Informationen, Regeln und Recommender/-Module).
3. Für mindestens eine Zielplattform (Windows, Mac OS, Linux) muss eine Client-Komponente in Form einer Desktop-Anwendung zur Verfügung stehen.
4. Die Desktop-Anwendung muss sowohl Informationen in Echtzeit darstellen als auch Feedback vom Anwender entgegennehmen können
5. Es muss eine Komponente existieren, die in Form eines Information Brokers zwischen Sensoren, Desktop-Client und Informationsbasis kommuniziert.

6. Es muss für die drei verschiedenen Umgebungstypen jeweils mindestens ein Sensor entwickelt werden, der in der Lage ist, bestimmte Benutzeraktivitäten zu überwachen und an den Client weiterzureichen.
7. Es muss für mindestens eine Zielplattform eine API bereitgestellt werden, die zur Übertragung von Sensordaten durch beliebige Anwendungen genutzt werden kann.

Ergänzend sind folgende nicht-funktionale Anforderungen an den Prototyp zu stellen:

8. Alle wesentlichen Bestandteile des Konzeptes sollen durch den Prototyp abgebildet werden.
9. Die Kommunikation zwischen Sensoren und Desktop-Clients soll über Standardprotokolle geschehen, um die universelle Einsetzbarkeit von PERCOP nachzuweisen.
10. Der Prototyp muss erweiterbar sein, um nachzuweisen, dass ein PERCOP-Framework – wie gefordert – erweiterbar ist.

Im nächsten Abschnitt wird zu den Anforderungen die Architektur des Prototyps präsentiert. Die Architektur wurde entsprechend den identifizierten Anforderungen entworfen.

## 8.2 Gesamtarchitektur des Prototyps

Auf höchster Abstraktionsebene besteht die Architektur der Plattform aus den vier Hauptkomponenten *Sensoren*, *Desktop-Client*, *Geschäftslogik* und *Web-Portal*. Bei den *Sensoren* handelt es sich um plattformunabhängige Anwendungen<sup>101</sup>, die in der Lage sind, kontextrelevante Daten wie z. B. Benutzeraktionen, Anwendungsparameter, Ereignisse oder Einstellungen zu ermitteln und anschließend zum Desktop-Client zu übertragen.

Der *Desktop-Client* repräsentiert in Form einer Rich-Client-Anwendung die Schnittstelle zu den Benutzern. Er empfängt die Daten der unterschiedlichen Sensoren und interagiert unter Verwendung einer *Information-Broker*-Komponente mit der Geschäftslogik, um relevante Informationen zu erhalten und den Benutzern präsentieren zu können. Des Weiteren kann der Benutzer über die Anwendung die Qualität der erhaltenen Informationen bewerten.

Die *Geschäftslogik* ist die zentrale Komponente der Plattform. Sie ist unter anderem verantwortlich für die Benutzer- und Kontextverwaltung, insbesondere jedoch für die

---

<sup>101</sup> Die Sensoren können beispielsweise in Form von Komponenten, Diensten oder Add-Ins realisiert werden.

Ausführung unterschiedlicher Information-Retrieval-Algorithmen zur Informationsermittlung. Über den Desktop-Client (bzw. durch die integrierte Information-Broker-Komponente) erhält die Geschäftslogik Informationen über die aktuelle Situation. Die Situationsdaten werden von der Geschäftslogik verwendet, um relevante Informationsartefakte zu ermitteln und in Form einer Ergebnisliste wiederum an den Client zurück zu senden.

Aufgaben des *Web-Portals* sind der Zugang und die Integration von Funktionen zur kollektiven Nutzung. Dazu zählen beispielsweise die Verwaltung und Bereitstellung von Umgebungen, Sensoren, Regeln, Recommendern, Funktionen und Informationen. Abbildung 75 zeigt die Gesamtarchitektur, welche in den folgenden Abschnitten detaillierter betrachtet wird.

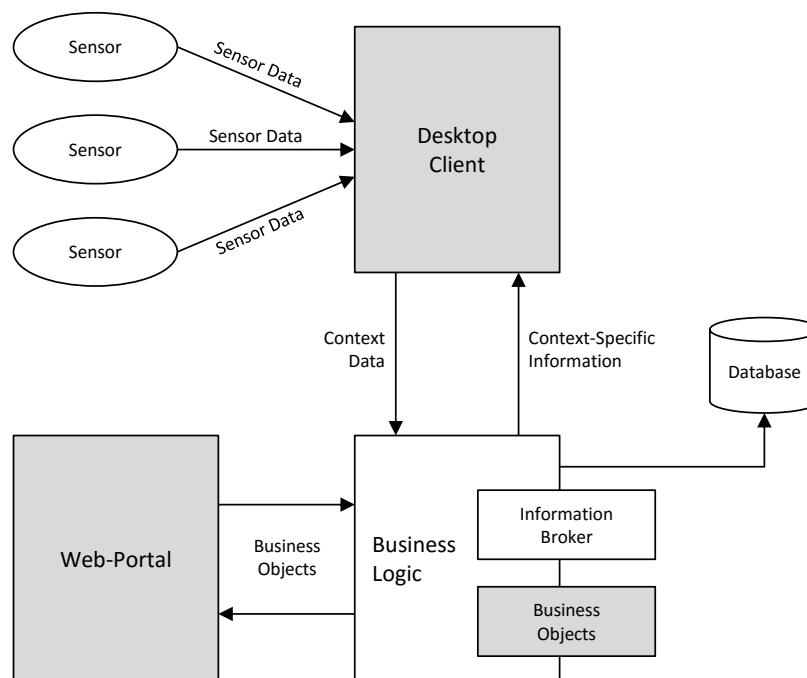


Abbildung 75: Kernelemente des PERCOP-Prototyps

### 8.3 PERCOP-Geschäftsobjekte

Geschäftsobjekte (*Business Objects*) repräsentieren die Entitäten, die zur Darstellung der Kontexte und Strukturen im Rahmen der Geschäftslogik für den Information-Retrieval-Prozess benötigt werden. Sie sind in einer zentralen Datenbank zu persistieren und durch die Geschäftslogik zu verwalten.



### 8.3.1 Kontextbasierte Geschäftsobjekte

Um einen Kontext abbilden zu können, werden die folgenden vier Objekttypen benötigt:

- *Environments*: Environment-Objekte repräsentieren Software-Anwendungen, mit denen Benutzer arbeiten. Auch Betriebssysteme sowie Kombinationen von Anwendungen können relevante Environment-Objekte sein (vgl. Abschnitt 2.1).
- *Sensor Data Types*: Sensoren sollen unterschiedliche Arten von Daten erfassen. Beispiele für *Sensor Data Types* im Kontext von Softwareanwendungen sind Fehlermeldungen, Fensterbeschriftungen oder die CPU-Auslastung.
- *Sensor Data Values*: Die konkreten Werte eines *Sensor Data Types* werden in den Objekten *Sensor Data Values* umgesetzt. Beispielsweise kann der *Sensor Data Type* „Fensterbeschriftung“ den *Sensor Data Value* „Neues Dokument“ besitzen oder der *Sensor Data Type* „CPU-Auslastung“ den *Sensor Data Value* „80%“.
- *Context*: Fasst die zuvor erläuterten Objekte zu einem Kontext-Objekt zusammen.

Mit Hilfe der drei *Context Business Objects* können (einzelne) Kontexte abgebildet werden. Ein Kontext wird somit beschrieben als ein konkreter Wert eines konkreten Sensors, der innerhalb einer konkreten Umgebung (Environment) diesen Wert ermittelt hat. Ein Beispiel wäre: „Benutzer erhält die Nachricht (*Sensor Data Value*) ‚Datei nicht gefunden‘ des *Sensor Data Types* ‚Fehlermeldung‘ in der Umgebung (*Environment*) ‚Adobe Photoshop‘“.

### 8.3.2 Allgemeine Geschäftsobjekte

Zusätzlich zu den kontextbasierten Geschäftsobjekten (*Context Business Objects*) ergänzen die folgenden allgemeinen Geschäftsobjekte die Architektur:

- *Functionalities*: Functionality-Objekte sind umgebungsspezifische Funktionalitäten, die von Anwendern genutzt werden.
- *Information*: Informations-Objekte repräsentieren jede Form von Informationsartefakt, welches für einen Benutzer in einer konkreten Situation nützlich sein kann.
- *Rules*: Die Verknüpfung von Informationen und Funktionen zu Kontexten wird mit Hilfe der Rule-Objekte realisiert. Die Komposition mehrerer Regeln wird durch die Rule-Operator-Objekte ermöglicht.
- *Users*: User-Objekte repräsentieren (registrierte) Anwender des Systems. Dies ist notwendig und sinnvoll, um benutzerspezifische Informationen zu ermitteln. Des

Weiteren können Benutzer unterschiedliche Rollen besitzen, die im Rahmen der Informationssuche berücksichtigt werden können.

- *Recommendations*: Bei Recommendations handelt es sich um positive oder negative Bewertungen, die ein Benutzer zu einer erhaltenen Information innerhalb eines Kontextes abgegeben hat. Diese können innerhalb der Empfehlungs-algorithmen berücksichtigt werden.

Abbildung 76 zeigt die interne Struktur der PERCOP-Geschäftsobjekte beim Prototyp.

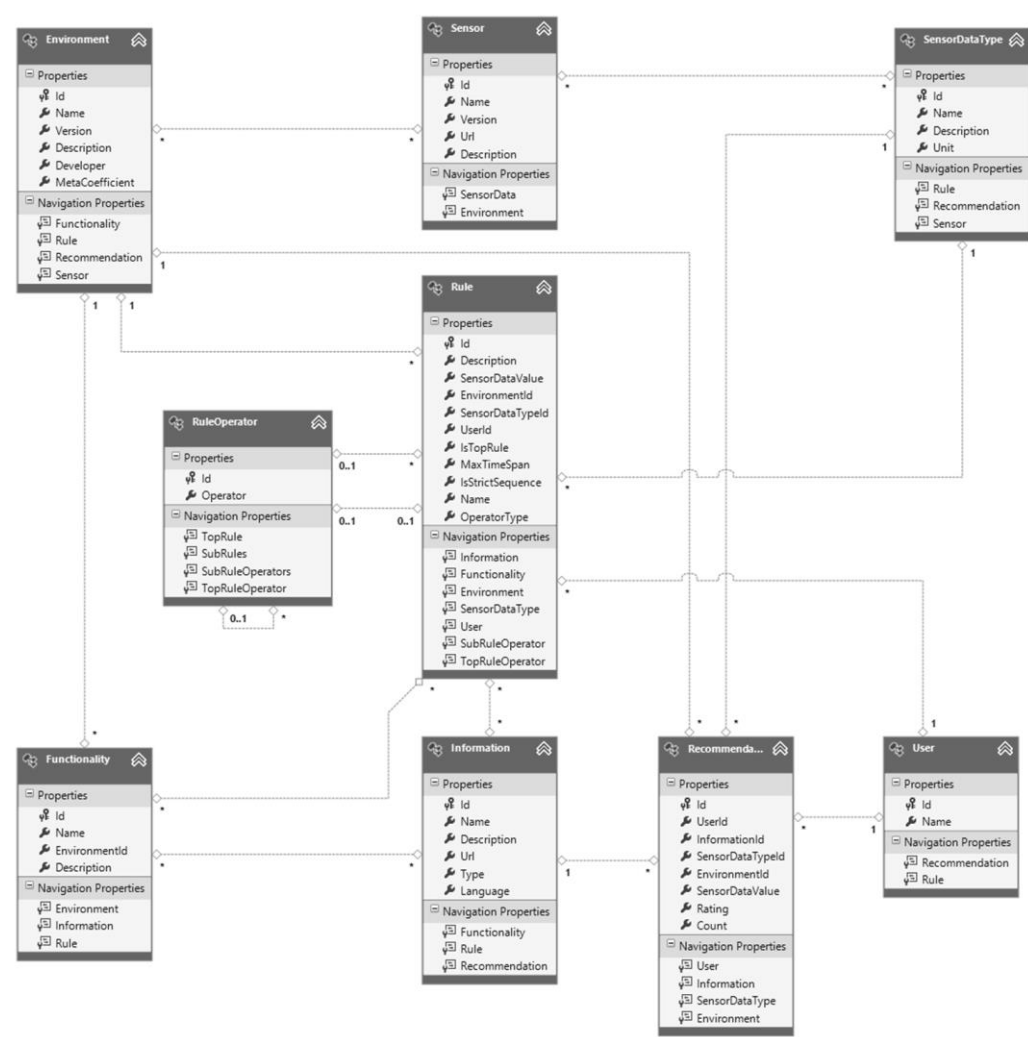


Abbildung 76: PERCOP-Geschäftsobjekte

## 8.4 PERCOP-Geschäftslogik

Die Komponenten auf der Geschäftslogik-Schicht (Business Logic) stellen die Kernfunktionalität des Prototyps dar. Für die PERCOP-Geschäftslogik werden drei Hauptkomponenten benötigt (siehe auch Abbildung 77):

- *UserAuthenticator*: Diese Komponente ist erforderlich für die Benutzer-Authentifizierung und -Zuordnung.
- *ContextManager*: Die ContextManager-Komponente soll die Kontextdaten von der Client-Anwendung empfangen, die Datenbank nach Geschäftsobjekten durchsuchen sowie eine Historie der Kontexte für jeden angemeldeten Benutzer vorhalten, um diese beispielsweise im Rahmen der Erstellung von Regeln nutzen zu können.
- *InformationBroker*: Die InformationBroker-Komponente soll eine Schnittstelle für die Client-Anwendung bereitstellen und den Information-Retrieval-Prozess initiieren. Die Komponente muss dazu den Benutzerkontext aus dem ContextManager analysieren, das interne Information-Retrieval-System einsetzen und dessen Ergebnisse (Information Results) zum Desktop-Client transferieren.

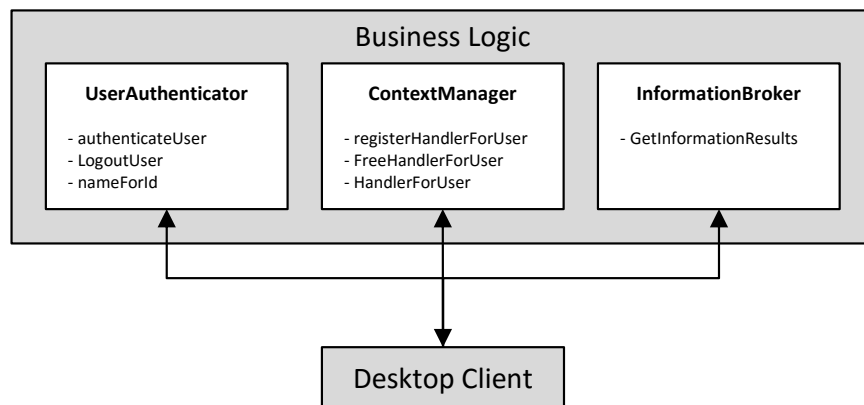


Abbildung 77: PERCOP-Geschäftslogik

### 8.4.1 Empfehlungssystem zur Informationsermittlung

Wie in Kapitel 6 erläutert, existieren für Empfehlungssysteme eine Reihe von Ansätzen und Verfahren, die jeweils unterschiedliche Einsatzgebiete und Ziele adressieren. Die einzelnen *Recommender*-Module des Frameworks sollen in Form von Programm-Bibliotheken auf der Plattform bereitgestellt werden und anschließend im Rahmen von zusammengestellten *Recommender-Konfigurationen* durch die Client-Anwendung und den Information-Broker nutzbar sein.

Im Rahmen des Prototyps wurden (exemplarisch) die folgenden Empfehlungsmodule implementiert. Die Auswahl der Module entstand mit Blick auf die beschriebenen Szenarios und Fallbeispiele.

Empfehlungsmodule sind im Kontext des Prototyps alle Komponenten, die auf Basis einer Eingabe (z. B. Kontext-Informationen, Suchabfragen oder Regeln) eine Liste von Informationsartefakten ermitteln und zurückliefern:

- *IRContextWebSearch*: Zentrales Modul zur Suche mit Hilfe bestehender Websuchmaschinen (zurzeit Google<sup>102</sup> und Bing<sup>103</sup>). Basiert auf Suchanfragen, die auf Basis der Kontextdaten automatisch erstellt wurden.
- *IRCollaborativeInformationBasedkNN*: Ermittelt relevante Informationen aus der BusinessObjects-Datenbank basierend auf kollaborativen element-basierten Empfehlungsalgorithmen.
- *IRCollaborativeUserBasedkNN*: Ermittelt relevante Informationen aus der BusinessObjects-Datenbank basierend auf kollaborativen benutzer-basierten Empfehlungsalgorithmen.
- *IRContentBasedInformationBasedkNN*: Ermittelt relevante Informationen aus der BusinessObjects-Datenbank basierend auf inhalts-basierten Empfehlungsalgorithmen.
- *IRDirectRecommendationRetrieval*: Ermittelt relevante Informationen aus der BusinessObjects-Datenbank, die direkt einem konkreten Kontext zugeordnet werden können.
- *IRRuleInformationRetrieval*: Dieses Modul wertet die durch Benutzer erstellten Regeln aus und überprüft, ob diese den aktuellen Kontext betreffen. Falls dies der Fall ist, werden Informationen ermittelt, die diesen Regeln zugeordnet sind.
- *IRRuleFunctionalityWebSearch*: Ermöglicht die Informationssuche im Web basierend auf Suchabfragen, die aufgrund der durchgeführten Funktionalität erstellt werden. Auch dieses Modul wertet dazu erstellte Regeln aus.
- *IRStackOverflowRetrieval*: Das Modul ist in der Lage, die StackOverflow<sup>104</sup>-Community nach kontextrelevanten Themen zu durchsuchen.
- *IRTwitterRetrieval*: Das Modul ist in der Lage, Twitter<sup>105</sup> nach kontextrelevanten Meldungen zu durchsuchen.
- *IRYahooAnswersRetrieval*: Das Modul durchsucht Yahoo Answers<sup>106</sup> nach kontextrelevanten Informationen.

---

<sup>102</sup> Siehe <http://www.google.de>

<sup>103</sup> Siehe <http://www.bing.com>

<sup>104</sup> Siehe <http://www.stackoverflow.com>

<sup>105</sup> Siehe <http://www.twitter.com>

<sup>106</sup> Siehe <http://www.yahooanswers.com>

Die zuvor beschriebenen Empfehlungsmodule liefern jeweils als Output eine Menge von Informationsartefakten. Zur Strukturierung und Bereinigung der Ergebnisse wurden (exemplarisch) folgende Module implementiert:

- *IRBasicDuplicateRemoval*: Entfernt Duplikate aus dem Ergebnis-Satz.
- *IRBasicWeightedRankParallelization*: Ermöglicht die parallele Ausführung mehrerer IR-Module. Jedes Modul wird aufgerufen und die Ergebnisse werden in ein Ergebnis zusammengefasst. Dabei werden Gewichtungen und Bewertungen berücksichtigt.
- *IRCascadingWeightedJoin*: Integriert das Ergebnis eines IR-Modules in die Liste vorheriger Ergebnisse.
- *IRMetaDataRetrieval*: Ermöglicht die Anreicherung der Ergebnisse mit Metadaten, wie beispielweise die durchschnittliche Benutzerbewertung.

Abbildung 78 stellt die Elemente und Beziehungen der Geschäftslogik und -objekte in Hinblick auf die Informationsermittlung grafisch dar.

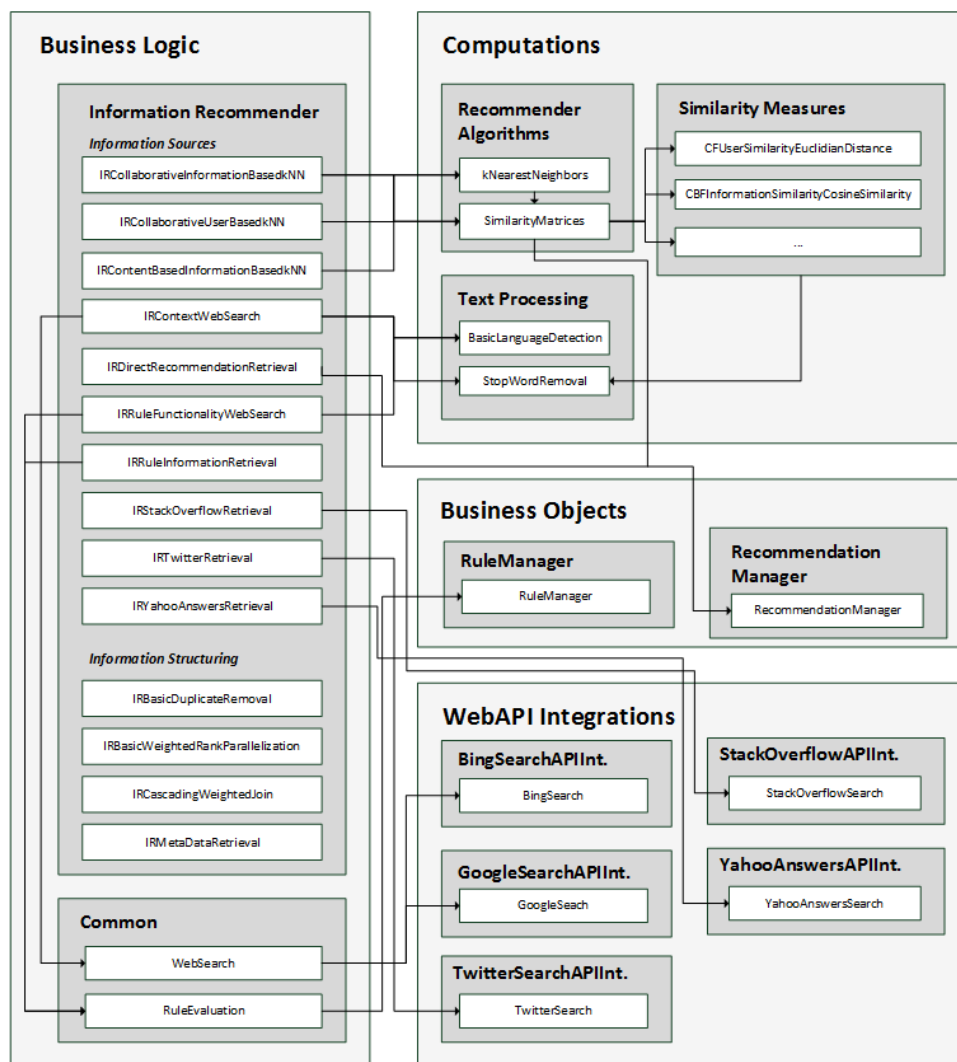


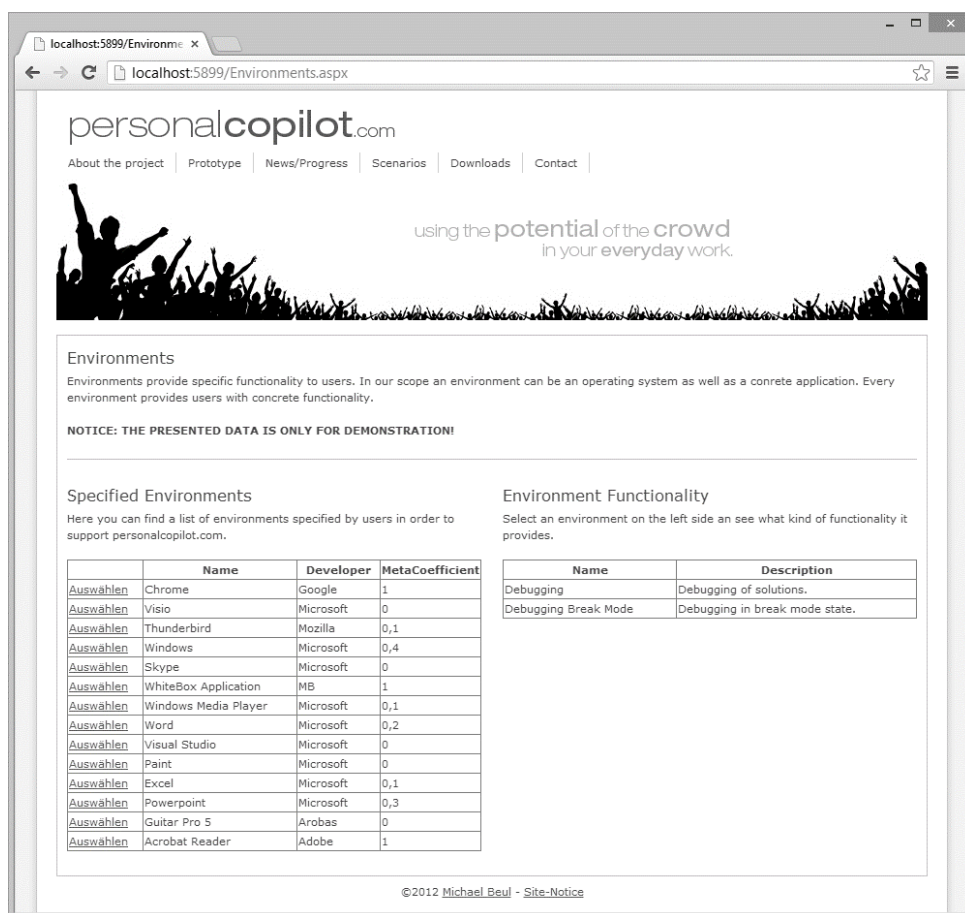
Abbildung 78: Beziehungen und Abhängigkeiten der Geschäftsobjekte und -logik

## 8.5 PERCOP-Web-Plattform

Die Web-Plattform soll die Schnittstelle zur Crowd bilden. Die in Form eines Portals konzipierte Plattform muss der breiten Masse dazu alle zuvor beschriebenen Funktionen bereitstellen, die von der Crowd genutzt werden sollen. Dazu zählen entsprechend der Ausführungen in den Abschnitten 3.2.2 und 5.6.4 die Verwaltung der Umgebungen, Funktionen, Informationen, Regeln, Sensoren und Empfehlungs-Module und -Konfigurationen; die entsprechenden Komponenten werden in den folgenden Abschnitten im Detail betrachtet.

## 8.5.1 Environment Manager

Innerhalb des Moduls *Environment Manager* können die unterschiedlichen Umgebungen verwaltet werden. Die Umgebungen stellen den Anknüpfungspunkt für die weiteren Module dar: Zu jeder Umgebung können Funktionalitäten spezifiziert werden, denen anschließend Informationen zugeordnet werden können. Mit Hilfe der Sensoren können konkrete Handlungen von Benutzern innerhalb der Umgebungen erkannt und bestimmten Funktionen und Informationen zugeordnet werden. Abbildung 79 zeigt die Oberfläche zur Verwaltung der Umgebungen.

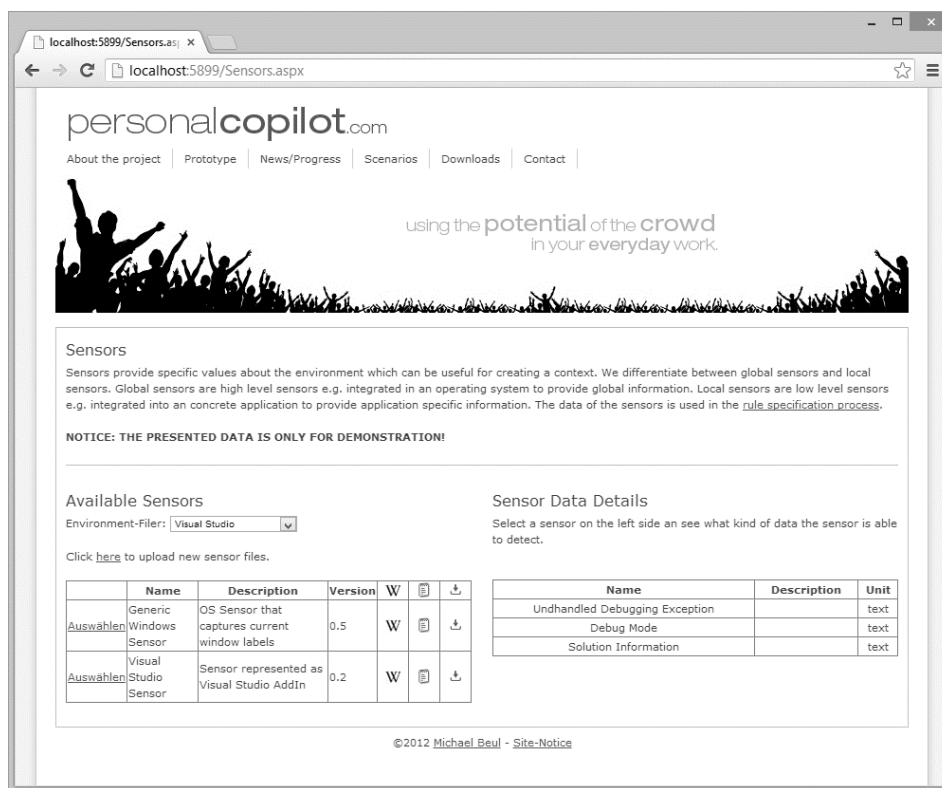


**Abbildung 79: Environment Manager des PERCOP-Prototyps**

Im Prototyp wurden die Umgebungsobjekte zunächst in vergleichsweise einfacher Form durch die Attribute Name, Beschreibung und Version abgebildet. Bei einer großen Anzahl verschiedener Umgebungen ist selbstverständlich eine Strukturierung und Versionierung erforderlich.

## 8.5.2 Sensor Manager

Das Modul *Sensor Manager* soll die Verwaltung der Sensoren ermöglichen. Über das Modul sollen virtuelle Sensoren auf die Plattform hochgeladen werden können. Die Sensoren bestehen aus zwei Hauptbestandteilen, zum einen dem Sensor in Form einer Komponente oder einer ausführbaren Datei und zum anderen den zugehörigen Metadaten. Die Metadaten spezifizieren, welche Daten der Sensor ermitteln kann und in welchem Format (Einheit und Wertebereich) diese Daten vorliegen. Zudem können auch Fuzzy-Logik-Werte in den Metadaten eingebettet werden, um die Erstellung komplexer Regeln zu ermöglichen.



**Abbildung 80: Sensor Manager des PERCOP-Prototyps**

Abbildung 80 zeigt die Verwaltungsoberfläche für Sensoren, wie sie für den Prototyp entworfen wurde. Zu jedem Sensor wird ein bzw. werden zwei Verweise angegeben: Über den Wiki-Verweis gelangt der Benutzer direkt auf eine entsprechende Inhaltsseite eines Wikis, in dem der Sensor beschrieben wird sowie auch ein Diskussionsforum bereitsteht. Wird der Sensor im Rahmen eines Open-Source-Projekts entwickelt, gelangt der Benutzer zudem über den Source-Verweis direkt auf eine entsprechende Kollaborationsplattform, über die er sich an der Entwicklung beteiligen kann.



### 8.5.3 Rule Manager

Die Erstellung von Regeln soll das Modul *Rule Manager* ermöglichen. Abbildung 81 zeigt einen Screenshot des prototypisch realisierten Moduls. Zunächst wählt der Benutzer eine Umgebung, anschließend einen Sensor und schließlich die konkreten Sensordaten. Die Daten können nun innerhalb der Regel-Formulierung verwendet werden und können durch Vergleiche (< > =) und Verknüpfungen (AND OR) logisch miteinander verknüpft werden. Die erstellte Regel kann anschließend direkt mit einer Funktionalität bzw. einer konkreten Information verknüpft werden.

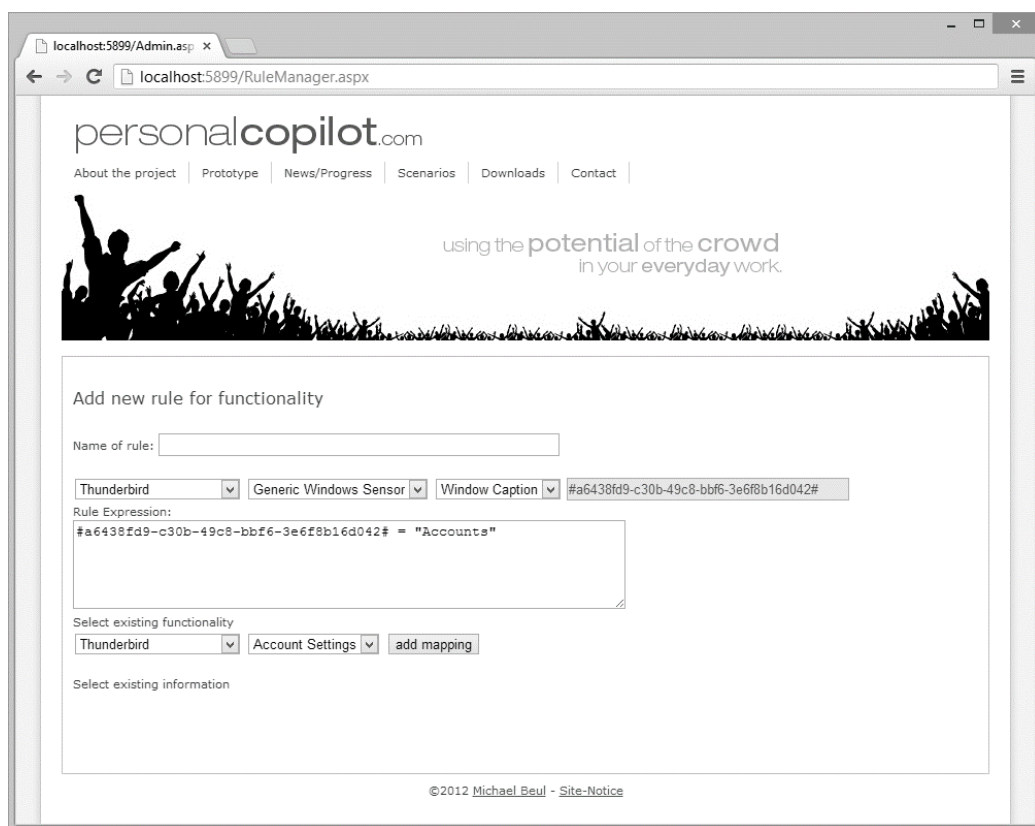


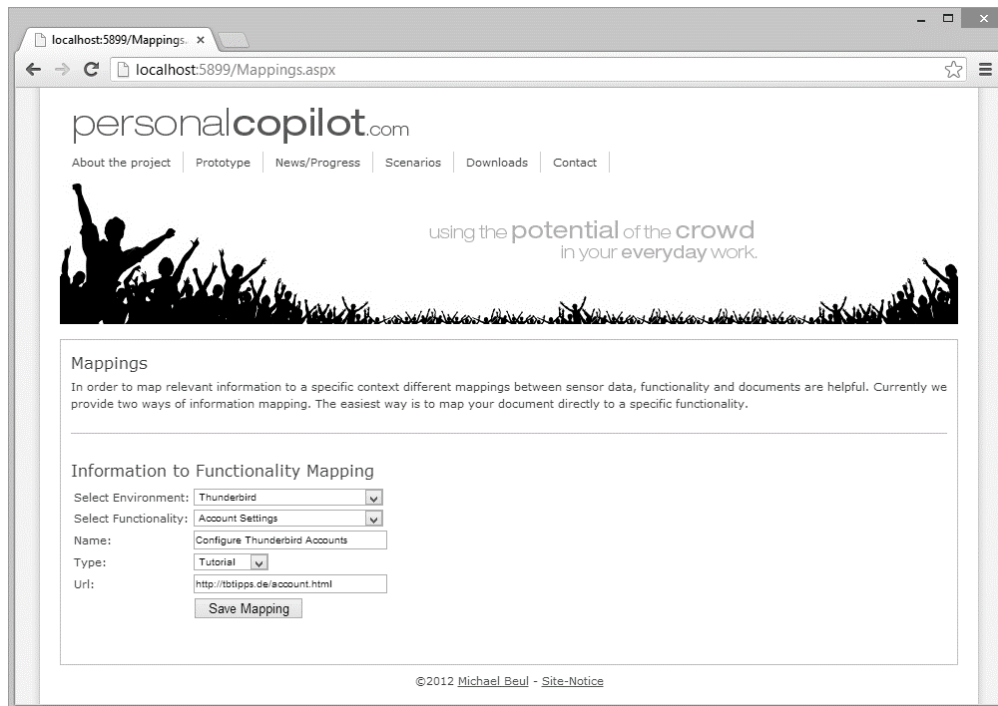
Abbildung 81: Rule Manager des PERCOP-Prototyps

Ergänzend zur Regelerstellung innerhalb des Web-Portals bietet der Regelassistent des Desktop-Clients (Abschnitt 8.7) einen erweiterten Funktionsumfang und eine weiterentwickelte Benutzerschnittstelle.

### 8.5.4 Direkte Zuordnungen (Mappings)

Das Modul *Mappings* soll unabhängig von den definierten Regeln die direkte Zuordnung von Informationsartefakten zu Funktionalitäten einer Umgebung ermöglichen. Abbildung

82 zeigt, wie den Benutzern nach Auswahl der jeweiligen Umgebung deren erfassten Funktionalitäten angezeigt werden. Nach Auswahl einer Funktionalität kann dann direkt ein konkretes Informationsartefakt in Form eines Links erstellt werden.



**Abbildung 82: Mapping-Modul des PERCOP-Prototyps**

Die direkten Zuordnungen können durch das Framework sehr schnell und effektiv ausgewertet werden, da keine Recommender-Module benötigt werden. Sobald ein Sensor die Nutzung einer definierten Funktionalität erfasst, werden zunächst die direkten Zuordnungen berücksichtigt, da diese im Sinne der Framework-Nutzung voraussichtlich eine sehr hohe Relevanz für den aktuellen Kontext besitzen.

## 8.6 Sensoren

Die Aufgabe der Sensoren ist die Ermittlung von Aktionen und Zuständen innerhalb der Umgebung. Sie sollen keine ergänzende „Intelligenz“ besitzen, damit eine feingranulare Aufteilung und die zielgerichtete Verteilung der Aufgaben in der Crowd erreicht werden. Somit können die spezifischen Fähigkeiten der Crowd-Mitglieder bestmöglich genutzt werden. Die Intelligenz soll deshalb erst durch die Verknüpfung mit Kontexten, Funktionen oder Informationen (z. B. innerhalb einer Regel) entstehen.

Technisch wurde diese Anforderung in der Form realisiert, dass die jeweiligen Sensoren über eine standardisierte Schnittstelle (RESTful-API) ihre *SensorID*, die konkrete *Sensor-DataId* und den aktuell ermittelten *Wert* an den Information Broker übermitteln. Diese Umsetzung ermöglicht eine einfache und plattformunabhängige Anbindung beliebiger Sensoren an die Plattform.

Im Prototyp wurden als Beispiele zwei Sensoren umgesetzt. Bezüglich der unterschiedenen Umgebungstypen ordnet sich einer der Sensoren in die BlackBox-Umgebung, der andere in die GrayBox-Umgebung ein.

Die Integration in eine WhiteBox-Umgebung bildet den einfachsten Fall, da bei ihr die Umgebung nicht überwacht werden muss, sondern aus der Umgebung heraus explizit die Benutzeraktivitäten gesendet werden. Dafür wurde im Rahmen des Prototyps eine API entwickelt, die in die Entwicklung von Anwendungen eingebunden werden kann.

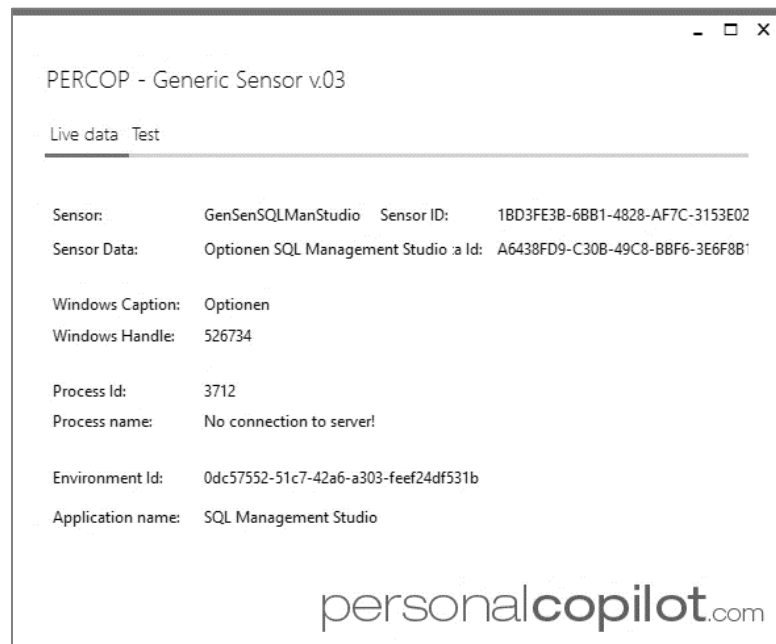
### 8.6.1 Generischer Sensor

Der generische Sensor soll Sensordaten ermitteln, die außerhalb der konkreten Anwendungen liegen bzw. anwendungsübergreifend von Interesse sind. Dabei handelt es sich beispielsweise um Informationen, die in Echtzeit über Betriebssystemfunktionalität ermittelt werden können. Im Rahmen des Prototyps wurde ein generischer Sensor entwickelt, der mit Hilfe der API des Betriebssystems unter anderem folgende Werte ermitteln kann:

- CPU- und Hauptspeicher-Auslastung,
- Massenspeicher-Verwendung,
- laufende Prozesse/Dienste sowie
- aktive Anwendung und aktuelles Fenster im Vordergrund.

Insbesondere mit Hilfe der Anwendungs- und Fenstererkennung ist es möglich, nützliche Kontextinformationen aus Anwendungen zu ermitteln, die keinen Zugriff auf Programmfunktionalität gewähren. Somit ermöglicht der generische Sensor die Nutzung des Frameworks auch innerhalb einer BlackBox-Umgebung.

Abbildung 83 zeigt die Statusanzeige der generischen Sensor-Applikation. Die Anwendung ist so konzipiert, dass sie unauffällig (nicht sichtbar für den Benutzer) im Hintergrund ausgeführt werden kann. Durch die Statusanzeige kann der Nutzer jedoch auch zu jeder Zeit sehen, welche Informationen aktuell gesammelt und zur Informationssuche genutzt werden. Die Transparenz soll zum einen die Akzeptanz der Nutzung erhöhen, zum anderen zur aktiven Beteiligung an der Erweiterung der Sensoren motivieren.



**Abbildung 83: Generischer Sensor des PERCOP-Prototyps**

## 8.6.2 Integrierter Sensor

Zur Demonstration der Realisierung eines Sensors innerhalb einer GrayBox-Umgebung wurde ein Sensor entwickelt, der mit Hilfe einer Add-In-Technologie die Integration in bestehende Anwendungen ermöglicht. Im konkreten Fall wurde für die Entwicklungsumgebung *Visual Studio* der Firma Microsoft ein Add-In entwickelt. Das prototypisch realisierte Add-In ist in der Lage, Fehlermeldungen des Compilers sowie Fehler während der Programmausführung zu erkennen und diese zur Information-Broker-Komponente zu übertragen.

Des Weiteren können Informationen über das Projekt ermittelt werden, beispielsweise die Struktur und die Verwendung von Fremdkomponenten. Mit Hilfe der von dem Add-In gelieferten Informationen kann automatisch nach Hilfestellungen gesucht werden. Beispielsweise kann eine spezielle Recommender-Konfiguration (hier der *IRStackOverflow-Recommender*) eingesetzt werden, um direkt nach konkreten Antworten bzgl. der Fehlermeldung auf spezialisierten Plattformen für Softwareentwicklung zu suchen. Diese Möglichkeit wird unten innerhalb eines Fallbeispiels (siehe Abschnitt 9.2) aufgegriffen.

Abbildung 84 zeigt das Add-In im Einsatz sowie auch die Nutzung der PERCOP-API bei der Entwicklung einer Anwendung (WhiteBox-Umgebung).

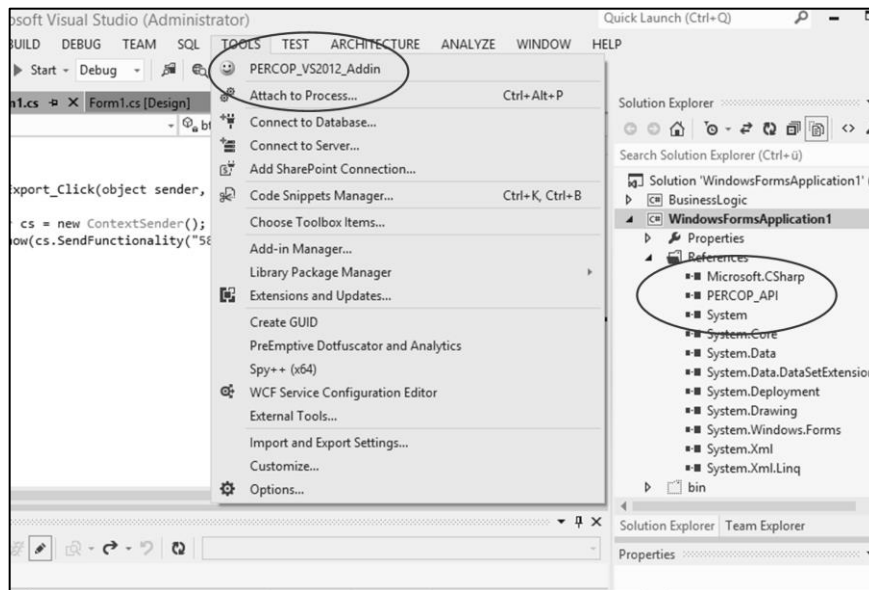


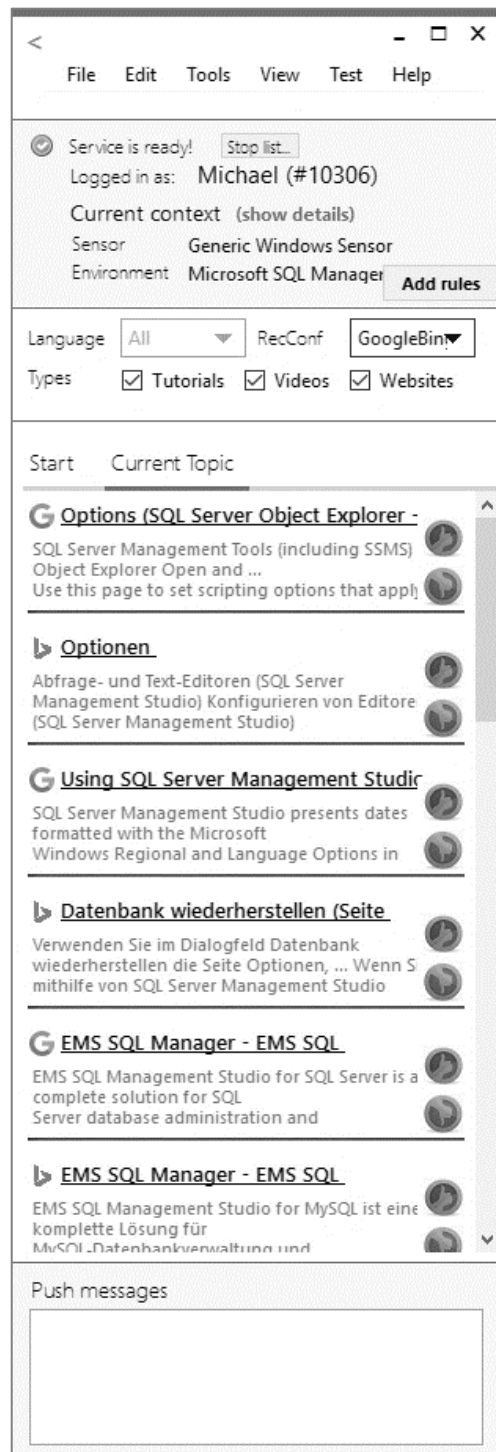
Abbildung 84: Visual Studio-AddIn und API-Nutzung

## 8.7 Desktop-Client

Der Desktop-Client soll die Schnittstelle zum Benutzer in Form einer Rich-Client-Anwendung realisieren. Sie ist so konzipiert und realisiert, dass sie dauerhaft auf dem Bildschirm sichtbar ist, während der Benutzer seiner Arbeit nachgeht. Werden relevante Informationen ermittelt, stellt der Client sie getrennt nach den jeweiligen Umgebungen in Form einer Liste dar. Die Darstellung erfolgt in Echtzeit, beeinträchtigt dabei nicht die Arbeitsumgebung der Benutzer. Nimmt der Benutzer eine Information wahr, die ihm interessant erscheint, kann er diese auswählen und wird automatisch zur jeweiligen Quelle weitergeleitet.

Der Benutzer kann zu jeder Zeit einsehen, welche Informationen zur Suche verwendet wurden und warum ihm bestimmte Informationen empfohlen werden. Des Weiteren kann er auf einfache Art und Weise die Relevanz der empfohlenen Artefakte bewerten<sup>107</sup>. Abbildung 85 zeigt den prototypisch realisierten Desktop-Client. Wie in der Abbildung zu sehen ist, wurde eine Recommender-Konfiguration (RecConf) „GoogleBing“ gewählt. Die Ergebnisliste zeigt daraufhin Elemente, die zunächst durch einen Google- sowie einem Bing-Recommender ermittelt wurden und anschließend zu einer Gesamtliste aggregiert wurden.

<sup>107</sup> An dieser Stelle wurde bewusst auf ein mehrstufiges Bewertungsverfahren verzichtet und lediglich ein zweistufiges System implementiert. Der Benutzer kann entscheiden, ob die Information hilfreich war oder nicht. Ein mehrstufiges System würde zum einen keinen großen Nutzen bringen, zum anderen die Bereitschaft der Benutzer negativ beeinflussen.



**Abbildung 85: PERCOP Desktop-Client**

Um die Erstellung von Regeln möglichst einfach zu gestalten, wurde ein Regel-Assistent konzipiert und ebenfalls prototypisch umgesetzt. Er erlaubt dem Benutzer unter Verwendung seines Aktivitäten-Verlaufs auf einfache Art und Weise neue Regeln zu erstellen. Dazu kann er seine zuletzt durchgeführten Aktionen durch Verknüpfungen in Regeln

transformieren, und diesen Regeln dann direkt Funktionen und Informationen zuordnen. Abbildung 86 zeigt den prototypisch realisierten Reglassistent.

Wenn aus dem Aktivitätenverlauf mehr als eine Aktivität ausgewählt wird, können diese Aktivitäten durch verschiedene Operatoren miteinander verknüpft werden. Des Weiteren ist es möglich, festzulegen, nach welcher Zeitspanne eine Regel ausgeführt werden soll. Auch eine direkte Zuordnung von konkreten Informationsartefakten oder ausgewählten Funktionalitäten einer Umgebung ist bei der Erstellung einer Regel möglich.

**Specify Your Rule**  
You are about to add a rule for the context to the right. You may associate it with a specific information or with a functionality of the respective environment. Please make your choice and enter your rule details.

☒ **Create Rule**

**Rule Options**

Name:

Operator:  ☐ Strict Sequence

Maximal Timespan: Minutes  Seconds

**Your selected context**

Context Count:

Show No.:

Environment:

Sensor Data Type:

Data Value:

Arrange at the following position in rule sequence:

☐ **Create Rule + New Information**

An information represents a reference to a helpful resource. Select this option if you want to directly associate an information with your selected context.

Title:

URL:

Type:

Description:

Language:

☐ **Create Rule + Link With Functionality**

A functionality represents a certain function or working scenario in an environment, which may be linked with helpful information later on. Select this option, if you wish to associate your selected context with an existing or a new functionality.

Environment:

☒ **Existing Functionality**

Name:

☐ **New Functionality**

Name:

Description:

**Abbildung 86: Reglassistent des PERCOP-Prototyps**

## 8.8 PERCOP-Gesamtprozess

Im Folgenden wird der Prozess beschrieben, der zu durchlaufen ist, um zu einer konkreten Situation eines Benutzers eine Liste relevanter Informationen zu ermitteln und ihm zu präsentieren. Der Fokus der Beschreibung liegt auf der Betrachtung der jeweiligen Elemente des Prototyps.

1. Ein Sensor erkennt eine Aktivität des Benutzers. Sofort übermittelt er die folgenden Daten zum Desktop-Client:
  - *EnvironmentId* (Eindeutige ID der Umgebung),
  - *SensorId* (Eindeutige ID des Sensors),
  - *SensorDataTypeId* (eindeutige ID des SensorsDatentyps),
  - *SensorDataValue* (gemessener Wert),
  - *Timestamp* (aktueller Zeitstempel) und optional
  - *FunctionalityId* (eindeutige ID einer konkreten Funktionalität).
2. Der Client empfängt die Daten, registriert diese und reicht sie direkt weiter zur ContextManager-Komponente, die wiederum die Daten zum ContextHandler des aktuellen Benutzers sendet.
3. Der ContextHandler überprüft die Daten, gleicht sie mit der Geschäftsobjekt-Datenbank ab und speichert das erstellte Kontextobjekt in die ContextQueue des jeweiligen Benutzers.
4. Der Client wird informiert, dass ein gültiger Kontext registriert wurde. Daraufhin ermittelt der Client die entsprechenden Objekte und stellt eine Anfrage an den InformationBroker.
5. Der InformationBroker empfängt die Anfrage und reicht diese an die aktivierte Recommender-Konfiguration weiter.
6. Innerhalb der Recommender-Konfiguration werden verschiedene Information-Recommender-Module aufgerufen, von denen jeder seinen InformationRetrieval- und Verarbeitungsprozess ausführt. Während der Ausführung werden unterschiedliche Elemente verwendet (z. B. Verwaltungsklassen der Regeln und Empfehlungen sowie Web-APIs, Empfehlungsalgorithmen und Text-Manipulationen).
7. Die Liste der Ergebnisse wird von der Recommender-Konfiguration an den InformationBroker gesendet und an den Desktop-Client weitergeleitet.
8. Der Desktop-Client aktualisiert die Ansicht und stellt die Ergebnisliste in Form von ResultViewItems dar.

Abbildung 87 stellt den Gesamtprozess graphisch dar.



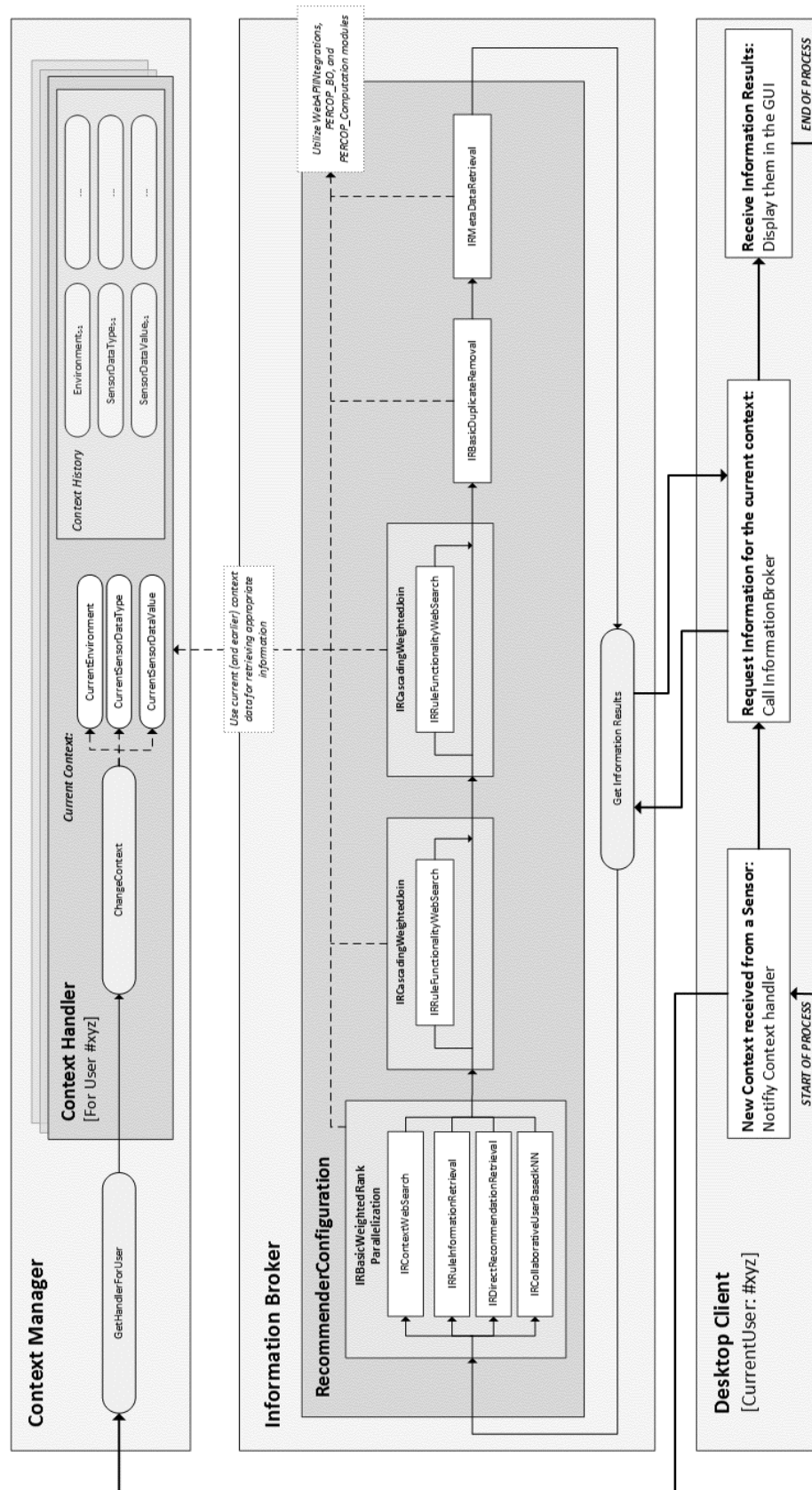


Abbildung 87: PERCOP-Gesamtprozess

## 8.9 Zusammenfassung

In diesem Kapitel wurde ein Prototyp vorgestellt, der ein PERCOP-Framework konkret implementiert und die Anforderungen des Frameworks erfüllt. Die wesentlichen Bestandteile des Prototyps sind der *Desktop-Client*, die *Sensoren*, das *Web-Portal* und die *Geschäftslogik*-Komponenten.

Der Desktop-Client dient im Wesentlichen der *Informationsdarstellung*. Die Sensoren ermitteln Umgebungszustände und Benutzeraktivitäten. Die Geschäftslogik-Komponenten erstellen aus den ermittelten Daten Kontextobjekte und nutzen diese mit Hilfe von Recommender-Konfigurationen zur *Informationsermittlung*. Aufgabe des Web-Portals ist die Integration der *Crowd*. Das Portal bietet eine Vielzahl von *Beteiligungsmöglichkeiten* und ermöglicht zudem die Förderung der unterschiedlichen Motivationsaspekte.

Die technische Umsetzung in Form eines konkreten Prototyps zeigt, dass die Realisierung der erarbeiteten Konzepte nicht nur theoretisch, sondern auch technisch/praktisch möglich ist. Zudem weist der Prototyp bzw. seine Entwicklung nach, dass für die konkrete Implementierung eines PERCOP-Frameworks keine wesentlichen Einschränkungen bestehen.

Einige Aspekte des Frameworks wurden im Rahmen des Prototyps nicht vollständig implementiert. Dies betrifft insbesondere die Verwaltungsmöglichkeiten auf der Web-Plattform durch die Crowd. Beispielsweise wurden Recommender-Konfigurationen zunächst fest implementiert. Auch die Rollen der Benutzer wurden bei der Informationssuche nicht technisch umgesetzt. Diese Einschränkungen haben jedoch keinen negativen Einfluss auf die Evaluation des Frameworks durch den Prototyp, da die technische Umsetzbarkeit der nicht realisierten Funktionen als gegeben zu bewerten ist, da entsprechende Funktionalitäten in anderen Anwendungssystemen bereits umgesetzt wurden und somit nicht nachgewiesen werden müssen.

Zur weiteren Validierung wird der Prototyp im folgenden Kapitel innerhalb ausgewählter Fallbeispiele eingesetzt, um auch die Anwendbarkeit in realen Anwendungsszenarios zu überprüfen.

## 9 Fallbeispiele

Zusätzlich zur Realisierung eines Prototyps wurden zur weiteren Validierung unterschiedliche Fallbeispiele durchgeführt. Die Fallbeispiele ermöglichen den Einsatz des Prototyps in realen praxisrelevanten Szenarios. In diesem Kapitel werden drei Fallbeispiele vorgestellt, die sowohl unterschiedliche Domänen adressieren als auch differenzierte Umgebungen abdecken.

Kapitel 9 gliedert sich wie folgt: Zunächst wird in Abschnitt 9.1 allgemein auf die Motivation und Anforderungen von Fallbeispielen im Kontext des PERCOP-Frameworks eingegangen. Anschließend wird ein Fallbeispiel aus der Domäne *Softwareentwicklung* vorgestellt (Abschnitt 9.2). Hier wird der Umgebungstyp *GrayBox* umgesetzt. Abschnitt 9.3 beschreibt die Integration des Konzepts in eine neu entwickelte Software-Anwendung, was einer *WhiteBox*-Umgebung entspricht. Abschließend wird in Abschnitt 9.4 eine *BlackBox*-Umgebung innerhalb der Domäne *Bildbearbeitung* betrachtet. Abschnitt 9.5 fasst die Erkenntnisse zusammen und betrachtet diese kritisch.

### 9.1 Motivation und Anforderungen

Die prototypische Entwicklung hat gezeigt, dass das Framework technisch realisierbar ist. Zur Validierung seiner Anwendbarkeit werden Fallbeispiele durchgeführt; ihr Inhalt ergibt sich aus den in Abschnitt 1.3 vorgestellten Szenarios. Weitere Ziele der Fallbeispiele bestehen darin,

- zu zeigen, dass eine Übertragbarkeit des Konzeptes auf unterschiedliche Domänen möglich ist,
- nachzuweisen, dass das vorgestellte Konzept praktikabel/anwendbar ist,
- Einschränkungen und Möglichkeiten der Erweiterung aufzudecken und
- erste quantitative Aussagen über die Zeitersparnis bei der Informationssuche innerhalb der unterschiedlichen Szenarios treffen zu können.

Die Fallbeispiele wurden so gewählt, dass der generische Aspekt und die Adaptierbarkeit des Frameworks demonstriert wird. Entsprechend unterschiedlich wurden Domäne, Benutzergruppen, Umgebungen, Informationsquellen und Prozesse ausgewählt. Des Weiteren sollen die Fallbeispiele auch Einschränkungen und Erweiterungsmöglichkeiten veranschaulichen. Die konkrete Beteiligung der Crowd wird innerhalb der Fallbeispiele lediglich simuliert. Da die Motivations- und Beteiligungsaspekte jedoch innerhalb dieser Arbeit logisch-argumentativ hergeleitet wurden, ist für die Beteiligung der Crowd keine weitere Validierung erforderlich.

## 9.2 Fallbeispiel 1: Softwareentwicklung

Im Rahmen dieses Fallbeispiels wird die Domäne Softwareentwicklung betrachtet. Aufgrund der schnellen Veränderung von Technologien, Konzepten und Werkzeugen ist der Informationsbedarf nach aktuellen Informationen in dieser Domäne besonders stark ausgeprägt.

### 9.2.1 Ausgangssituation

Die Ausgangssituation des Fallbeispiels wird im Wesentlichen durch die verwendete Entwicklungsumgebung geprägt, in diesem Fall die Umgebung *Visual Studio (VS)* der Firma Microsoft. Adressierte Benutzer sind entsprechend Entwickler als „Informationsnachfrager“ beim Einsatz von Visual Studio zur Softwareentwicklung.

Des Weiteren existieren unterschiedliche (Web)-Expertensysteme wie beispielsweise die oben angesprochene Q&A-Plattform *Stack Overflow*. Die große Anzahl an Beiträgen und die hohe Anzahl von Benutzern auf dieser Plattform lassen darauf schließen, dass Softwareentwickler häufig diese Art von Expertensystemen zur Informationssuche nutzen.

*Stack Overflow* soll deshalb im Rahmen dieses Fallbeispiels als primäre Informationsquelle genutzt werden. Zudem sollen bestimmten Situationen mittels Regeln direkt konkrete Informationsartefakte zugeordnet werden. Über Add-Ins kann dazu direkt auf Funktionalitäten von Visual Studio zugegriffen werden.

### 9.2.2 Umsetzung mit PERCOP

Entsprechend der Ausgangssituation fokussiert das Fallbeispiel auf eine *GrayBox-Umgebung*, da Visual Studio die Möglichkeit bietet, über Add-Ins auf Funktionalitäten der Entwicklungsumgebung zuzugreifen. Entsprechend wurde ein *Sensor* in Form eines Add-Ins entwickelt, der folgende Informationen bzw. Zustände ermitteln und an den PERCOP-Client (Information Broker) übertragen kann:

- Informationen zur Umgebung (z. B. eingesetzte Version von Visual Studio),
- den Zustand der Erkennung des Auftretens einer Ausnahme (Exception),
- die Nachricht der aufgetretenen Ausnahme sowie
- die Struktur der VS-Projektmappe<sup>108</sup>.

---

<sup>108</sup> Software-Anwendungen und -Module werden in Visual Studio in Form von Projekten strukturiert, die wiederum innerhalb einer sog. Projektmappe (engl. Solution) gespeichert werden.

Des Weiteren wurde ein *Recommender-Modul* entwickelt, das die Q&A-Plattform *Stack Overflow* nach Inhalten durchsuchen kann; genutzt wird dazu die Programmierschnittstelle von *Stack Overflow*.

Die Anbindung von *Stack Overflow* über ihre Programmierschnittstelle ermöglicht eine Optimierung der Suche nach relevanten Inhalten. Beispielsweise können gezielt Beiträge bestimmter Autoren oder bestimmter Kategorien durchsucht werden. Das Recommender-Modul ist zudem in der Lage, die Beiträge zu filtern, die in Hinblick auf Antworten auf eine Frage als sog. *accepted solution* markiert wurden, also als Antwort, die die ursprüngliche Frage hinreichend beantwortet hat.

Insgesamt kann zum einen die Zeit der Suche deutlich beschleunigt werden, zum anderen kann die Relevanz der ermittelten und angezeigten Informationen erhöht werden, was dem Benutzer den Aufwand erspart, irrelevante Ergebnisse lesen und einschätzen zu müssen. Dies erhöht selbstverständlich die Benutzerakzeptanz.

Die Analyse der zugehörigen Projektstruktur in Visual Studio kann relevante Informationen bezüglich der Architektur und auch der Komplexität der Software liefern. Beispielsweise kann automatisch ermittelt werden, welche Komponenten innerhalb der Software eingesetzt werden. Diese Daten können wiederum genutzt werden, um konkret Informationsartefakte zu Komponenten oder Herstellern dieser Komponenten zu ermitteln und bereitzustellen.

Im Rahmen von Forschungsarbeiten des Lehrstuhls für Wirtschaftsinformatik und Softwaretechnik der Universität Duisburg-Essen werden in Hinblick auf Entscheidungsunterstützung bei der Softwareentwicklung die Verwendung von (Dritt-)Komponenten und deren Abhängigkeiten untersucht (vgl. Schwittek und Eicker 2012). Ein Ergebnis ist eine Plattform, die eine Übersicht über die unterschiedlichen Versionen von Komponenten bietet sowie auch über Beziehungen einer Komponente zu anderen Komponenten. Dadurch können unter anderem frühzeitig inkompatible Komponenten erkannt werden bzw. umgekehrt Kombinationen, die nach den bestehenden Erfahrungen konfliktfrei miteinander interagieren.

Auch diese Wissens-Plattform könnte durch ein PERCOP-Recommender-Modul abgefragt werden. Der Input in Form einer Liste der relevanten Komponenten würde durch einen Sensor (Visual Studio Add-In) ermittelt und an die Plattform übertragen werden. Der Benutzer würde dann auf Probleme hingewiesen werden, die bei der gewählten Komponenten-Kombination in der Vergangenheit aufgetreten sind.

Des Weiteren wurden mehrere Recommender-Konfigurationen erstellt, die Recommender-Module in unterschiedlicher Form miteinander verbinden. Dabei werden auch allgemeine Recommender-Module eingesetzt, die unter anderem Suchmaschinen abfragen. Ergänzend wurden passende Regeln zur Verknüpfung und Zuordnung spezifiziert.

Abbildung 88 visualisiert die einzelnen Bestandteile der konkreten Umsetzung von PERCOP in Fallbeispiel 1.

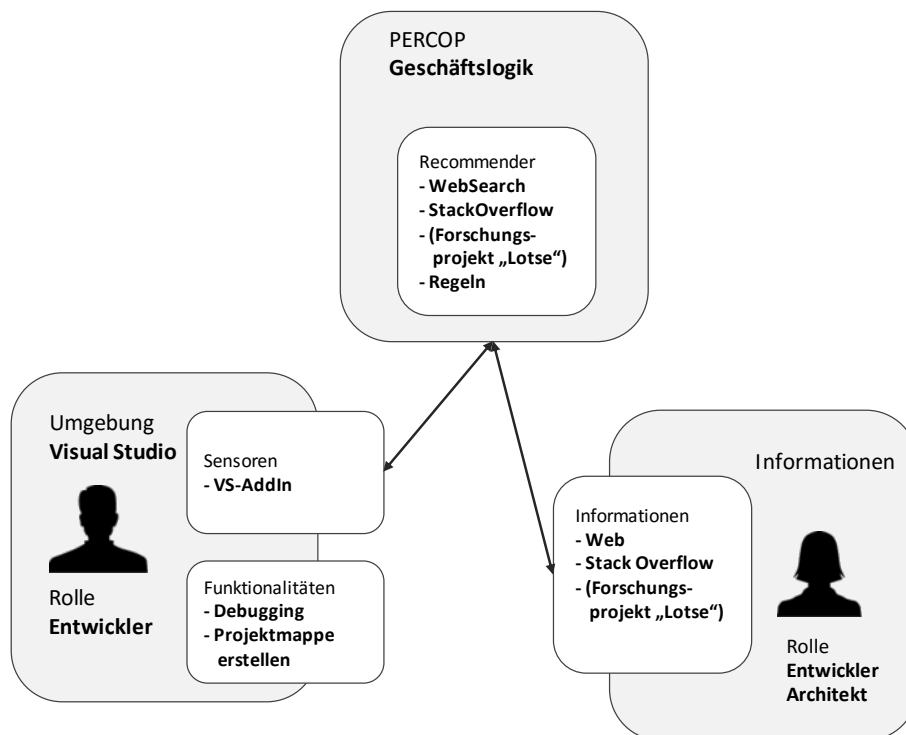


Abbildung 88: Übersicht zu Fallbeispiel 1

### 9.2.3 Erkenntnisgewinn

Das Fallbeispiel zeigt, dass in der Domäne Softwareentwicklung die Nutzung von PERCOP eine Automatisierung und Optimierung der Informationssuche ermöglicht. Die konkrete Anwendung dieses Fallbeispiels durch ausgewählte Testpersonen hat ergeben, dass die Nutzer (Softwareentwickler) während der Durchführung ihrer Arbeit nicht eingeschränkt bzw. abgelenkt wurden. Die Informationsbeschaffung konnte (nahezu) in Echtzeit realisiert werden. Auch die Zuordnung der Informationsquellen wurde positiv bewertet.

Das Fallbeispiel zeigt zudem, dass bereits bei einer vergleichsweise geringen Beteiligung der Crowd ein PERCOP-Framework hilfreiche Informationsartefakte ermitteln kann. Bei

der Anwendung des Fallbeispiels entstanden zudem zahlreiche Ideen für weitere Einsatzmöglichkeiten. Insbesondere die Analyse der Struktur des in der Entwicklungsumgebung geöffneten Projekts bildet ein großes zusätzliches Potenzial für weiterführende Informationen.

Erwartungsgemäß besitzen die direkt zugeordneten Informationsartefakte zu einem konkreten Kontext die höchste Relevanz, was natürlich abhängig von der Qualität der Informationen ist. Bei der Anwendung des Fallbeispiels fiel zudem auf, dass die visuelle Darstellung der Ergebnisse innerhalb der Liste dahingehend verändert werden sollte, dass die Herkunft und die Gewichtung der einzelnen Elemente optisch hervorgehoben werden.

## 9.3 Fallbeispiel 2: Verteilte Hilfe

Im zweiten Fallbeispiel soll gezeigt werden, dass eine hochdynamische und kontextsensitive Hilfefunktion einer Softwareanwendung durch die Implementierung des Frameworks realisiert werden kann. Es soll eine Möglichkeit geschaffen werden, bei der Hersteller einer Software sowohl die Erstellung als auch die Darstellung von anwendungsspezifischen Informationen und Hilfestellungen zur Nutzung zumindest zum Teil „auslagern“ können.

### 9.3.1 Ausgangssituation

Die Umgebung wird durch eine neu entwickelte Anwendung repräsentiert, bei deren Entwicklung die PERCOP-Funktionalität direkt bei der Entwicklung berücksichtigt und in den Programmcode integriert wird. Es liegt also eine *WhiteBox*-Umgebung vor. Bei der Anwendung handelt es sich beispielsweise um eine *RichClient*-Anwendung, welche auf Basis des .NET-Frameworks der Firma Microsoft entwickelt wird.

Die Informationen und Hilfetexte sollen aus einem Wiki bezogen werden, das zu den unterschiedlichen Funktionen der Anwendung die relevanten Informationen enthält. Des Weiteren sollen hilfreiche Video-Beiträge von öffentlichen Plattformen, beispielsweise aus Youtube, eingebunden werden.

### 9.3.2 Umsetzung mit PERCOP

Für die Integration von PERCOP direkt in eine Anwendung existieren grundsätzlich zwei Möglichkeiten: Da die Übertragung der Informationen zu dem Desktop-Client auf standardisierten Verfahren und Technologien basiert, kann die Integration entweder manuell

oder mit Hilfe einer Programmbibliothek erfolgen. Für das Fallbeispiel wurde die zweite Alternative gewählt und eine technologiespezifische Bibliothek entwickelt, die innerhalb der „eigenen“ Anwendung genutzt werden kann. Durch die Nutzung dieser Bibliothek kann eine sehr effiziente Integration erfolgen; so entspricht der Aufruf bzw. das Senden des jeweiligen Kontexts in etwa der Vorgehensweise des Loggens von Benutzeraktivitäten. Innerhalb des Programmcode werden beispielsweise mit der Anweisung „*PERCOP\_API.SendFunctionality(Umgebung,Funktionalität)*“ die nötigen Informationen zum Information Broker gesendet. Aufgrund der gewählten Technologie handelt es sich bei der Bibliothek um eine .Net-Assembly.

Auch der Zugriff auf das Wiki kann auf zwei Arten realisiert werden:

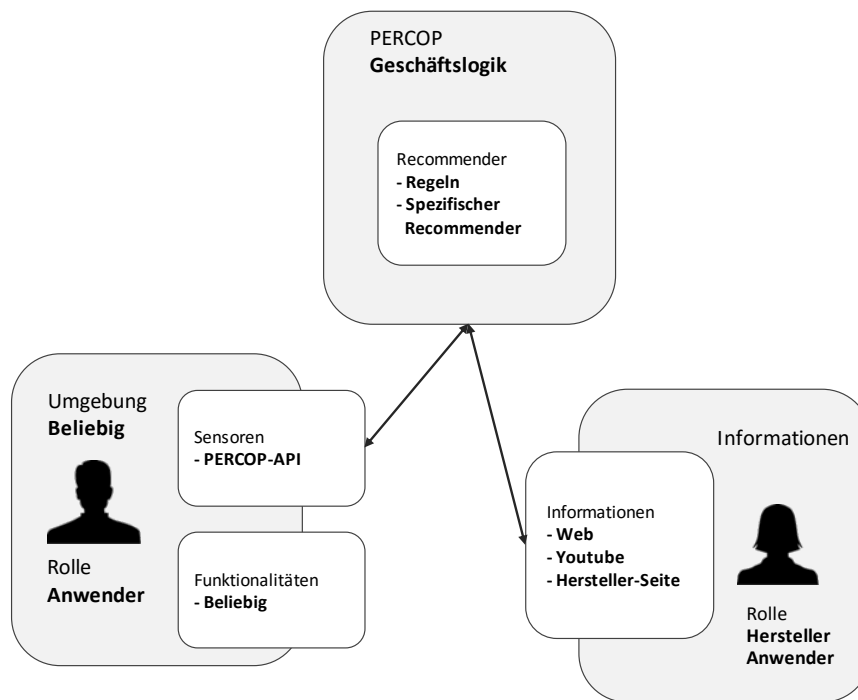
1. *Mit Hilfe von Regeln*: Da jeder Eintrag eines Wikis über eine eindeutige URL erreichbar ist, können Regeln definiert werden, die jeder Funktionalität der Anwendung die jeweilige Adresse des zugehörigen Artikels zuordnen.
2. *Mit Hilfe eines Recommender-Moduls*: Die Entwicklung und Verwendung eines (eigenen) Recommender-Moduls und einer Recommender-Konfiguration ermöglicht den direkten Zugriff auf das entsprechende Wiki. Dabei werden automatisch die passenden Einträge ermittelt und bereitgestellt.

Beide Alternativen besitzen Vor- und Nachteile: Bei der Verwendung von Regeln muss für jede Zuordnung von Funktionalität zu einer Informationsquelle eine Regel definiert werden. Dies kann bei komplexen Anwendungen mit großem Funktionsumfang sehr aufwändig und fehleranfällig werden. Dafür entfällt umgekehrt die Notwendigkeit der Entwicklung eines Recommender-Moduls. Ein Recommender-Modul hat wiederum den Vorteil, dass die Regelerstellung entfällt und zusätzliche Funktionalitäten des Wikis integriert werden können.

Die Zuordnung von Anwendungsfunktionalitäten zu nützlichen Video-Beiträgen kann durch direktes Mapping über eine Regel realisiert werden. Alternativ oder zusätzlich kann ein spezielles Recommender-Modul eingesetzt werden, das in der Lage ist, die Plattform YouTube selbständig zu durchsuchen.

Abbildung 89 visualisiert die einzelnen Bestandteile der konkreten Umsetzung von PERCOP in Fallbeispiel 2.





**Abbildung 89: Übersicht zu Fallbeispiel 2**

### 9.3.3 Erkenntnisgewinn

Das Fallbeispiel hat zum einen gezeigt, dass das Szenario der verteilten (dezentralen) Erstellung von Hilfe-Inhalten sehr gut durch ein PERCOP-Framework realisiert werden kann. Die unterschiedlichen Möglichkeiten zur Umsetzung bieten eine flexible Gestaltungsform.

Wie zu erwarten war, liefert die direkte Integration der Erkennung von Benutzeraktivitäten in die Anwendungslogik die besten Ergebnisse hinsichtlich der Kontexterkennung und -übermittlung, da die Nutzung von Sensoren (und die von ihnen zu ermittelnden Benutzer-Aktivitäten) entfällt. Die direkte Erkennung ist zum einen schneller, zum anderen können über sie auch Situationen definiert werden, die durch Sensoren möglicherweise nicht erfasst werden können (da die Anwendung beispielsweise keinen Zugriff auf benötigte Informationen erlaubt).

Die Geschwindigkeit der Informationsbereitstellung ist beim Einsatz eines eigenen Recommender-Moduls in Verbindung mit der direkten Integration der Kontexterkennung in die Anwendung am höchsten: Zum einen müssen keine Sensoren die Anwendung überwachen und Kontextänderungen erkennen und weiterleiten. Zum anderen entfällt die Auswertung einer möglicherweise großen Anzahl von Regeln.

## 9.4 Fallbeispiel 3: Bildbearbeitung

### 9.4.1 Ausgangssituation

Das dritte Szenario adressiert Umgebungen vom Typ *BlackBox*. In dem Fallbeispiel sollen einem Benutzer der Anwendung Photoshop von der Firma Adobe nützliche Informationen bzgl. seiner aktuellen Tätigkeit präsentiert werden<sup>109</sup>. Zum einen soll dem Benutzer eine konkrete Nachricht angezeigt werden, sobald dieser eine bestimmte Datei öffnet<sup>110</sup>. Zum anderen sollen im Internet verfügbare Artefakte ermittelt werden, sobald er bestimmte Funktionen der Anwendung nutzt.

### 9.4.2 Umsetzung mit PERCOP

Entsprechend der Ausgangssituation sind Sensoren notwendig, die „von außen“ das Verhalten der Anwendung und die Aktivitäten des Benutzers erfassen können. Im Rahmen des Prototyps wurde ein generischer Sensor entwickelt, der in der Lage ist, aktuelle Daten aus dem zugrundeliegenden Betriebssystem zu erfassen. Zu solchen Daten zählen neben Hardware-Eigenschaften auch geöffnete Fenster oder ausgewählte Menüeinträge. Da die Anwendung *Photoshop* sehr fensterorientiert arbeitet, konnten viele Funktionen und Zustände des Programms auf einfache Art und Weise durch die Erfassung der Fenster-Beschriftung zugeordnet werden<sup>111</sup>.

Die Informationssuche erfolgte im Rahmen des Fallbeispiels mit Hilfe eines *WebSearch-Recommender*-Moduls, welches auf Basis der Kontext-Informationen Suchabfragen für gängige Suchmaschinen erzeugt und ausführt. Zudem können Sprachen und Dokumenttyp konfiguriert werden.

Abbildung 90 visualisiert die einzelnen Bestandteile der konkreten Umsetzung von PERCOP in Fallbeispiel 3.

---

<sup>109</sup> Prinzipiell erlaubt die Software *Adobe Photoshop* auch die Entwicklung von Plug-Ins. Im Rahmen des Fallbeispiels wurde sie jedoch als *BlackBox*-Umgebung betrachtet und steht stellvertretend für eine beliebige Anwendung dieses Typs.

<sup>110</sup> Vorstellbar wären hier z.B. Kundeninformationen aus einem Firmen-Intranet zu dem zugehörigen Dokument oder Hinweise zu einer aktuelleren Version.

<sup>111</sup> In diesem Zusammenhang wurde festgestellt, dass bei sehr vielen Anwendungen, die auf Microsoft Windows Betriebssystemen ausgeführt werden, aufgrund der aktuellen Fensterbeschriftungen auf die aktuelle Tätigkeit des Anwenders bzw. Funktion der Anwendung geschlossen werden kann.

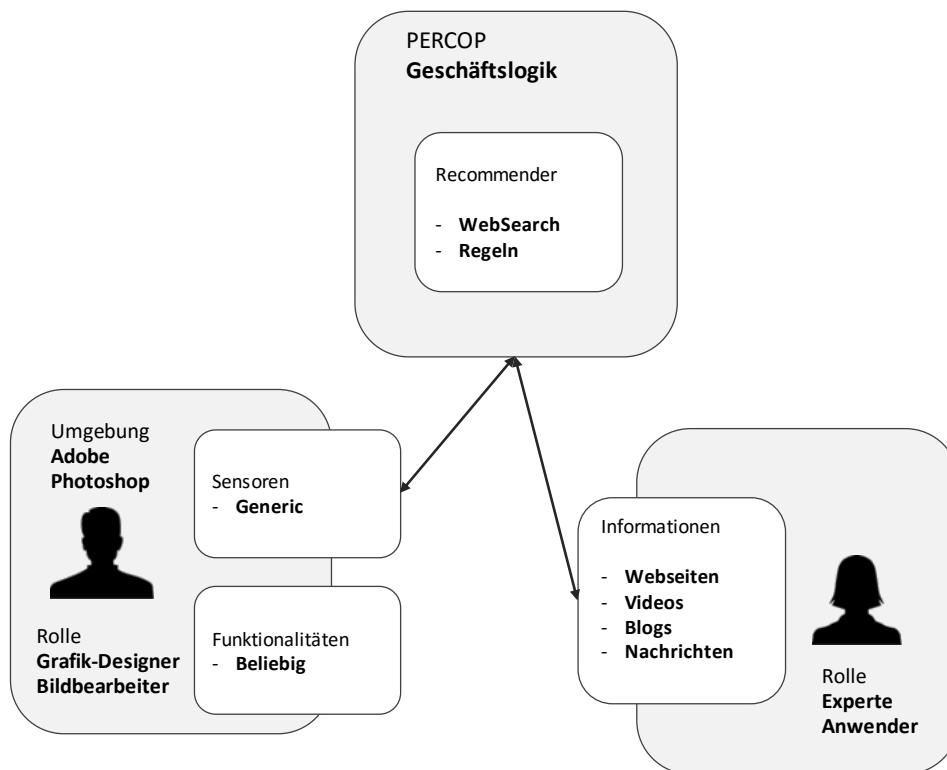


Abbildung 90: Übersicht zu Fallbeispiel 3

### 9.4.3 Erkenntnisgewinn

Bei der konkreten Anwendung des Fallbeispiels konnten (durch die Verwendung des generischen Sensors) den Funktionen der Anwendung viele Benutzeraktivitäten zugeordnet werden. Das Öffnen einer konkreten Datei wurde durch den Sensor unmittelbar erkannt und mit Hilfe einer Regel konnte die entsprechende Information dem Benutzer direkt angezeigt werden.

Während der Arbeit mit der Anwendung lieferte die Ermittlung von zugehörigen Informationen mit Hilfe der Web-Suche, sinnvolle, allerdings als allgemein einzustufende Ergebnisse. So wurde beispielsweise erkannt, dass der Benutzer einen bestimmten Filter ausgewählt hat (z. B. „Aquarell“), worauf hin eine Websuche allgemeine Informationen zu diesem Filter lieferte.

Um den Anwendern qualitativ hochwertigere Informationen liefern zu können, ist somit eine Ergänzung des im Rahmen des Prototyps erstellen (konkreten) Frameworks sinnvoll bzw. erforderlich. Dazu liefert das Fallbeispiel Ansätze für die Umsetzung: Der generische Sensor sollte zu einem anwendungsspezifischen Sensor erweitert werden, der beispielsweise mit Hilfe von Bilderkennungsfunktionalität ermittelt, um welchen Typ Bild es sich

bei dem aktuell geöffneten Dokument handelt (z. B. Portrait, Landschaft, Illustration, Farbe/Schwarz-Weiß), und diese Einstufung für die Informationssuche bereitstellt.

Ein weiteres Anwendungsbeispiel adressiert die Funktion der Stapelverarbeitung innerhalb von Photoshop. Diese ermöglicht es, wiederkehrende Aufgaben automatisch auszuführen. Kennt ein Benutzer diese Funktion nicht, würde er nicht danach suchen. Ein Sensor könnte jedoch registrieren, wenn ein Benutzer wiederholt die gleichen Aktivitäten manuell ausführt. Durch eine Regel könnte dem Benutzer dann konkret eine Information zu der Stapelverarbeitungs-Funktion präsentiert werden.

## 9.5 Zusammenfassung

Die im Rahmen der Arbeit durchgeführten Fallbeispiele decken unterschiedliche Domänen und Anwendungsszenarios ab. Dadurch konnten die Fallbeispiele zeigen, dass das Framework PERCOP in der Lage ist, die gestellten Anforderungen zu erfüllen. Die Fallbeispiele haben darüber hinaus ergeben, dass es sinnvoll sein kann, domänen-spezifische Ausprägungen des Frameworks zu erstellen und zu veröffentlichen. Diese könnten dann von interessierten Benutzergruppen in ihrem geschlossenen Bereich eingesetzt werden.

Des Weiteren wurde deutlich, dass die Darstellung der Ergebnisliste der Informationsartefakte noch optimiert werden sollte, damit der Benutzer „noch schneller“ Art, Inhalt und Herkunft der Informationen erfassen kann. Dies würde eine schnellere Bewertung durch den Benutzer ermöglichen und könnte somit den gesamten Informationssuchprozess beschleunigen.

Im Hinblick auf die Motivation der Beteiligung können die Fallbeispiele aufgrund der Größe der beteiligten Crowd lediglich Vermutungen liefern, die sicherlich nicht repräsentativ sind, sich jedoch in Hinblick auf die erarbeiteten Aspekte nach Einschätzung des Autors auf eine große Anzahl beteiligter Individuen übertragen lassen.

In den Fallbeispielen wurde beobachtet, dass unterschiedliche Benutzertypen unterschiedliche Beteiligungsinteressen besitzen – was sicherlich nicht überrascht. Es existierten sowohl Benutzer, die das Framework lediglich zum Informationskonsum nutzen, als auch Benutzer, die sich gerne bei der Regelerstellung, Bewertung und Programmierung engagieren.

# **TEIL IV**

## **Schlussbetrachtung und Fazit**

Der vierte Teil schließt die Arbeit mit einer Schlussbetrachtung, einer Zusammenfassung und einem Fazit ab. Insbesondere wird das entwickelte Framework in Hinblick auf die Forschungsmethode reflektiert und es werden sich anschließende Aktivitäten und Forschungsfragestellungen erläutert.

## 10 Schlussbetrachtung

Kapitel 10 schließt die Arbeit, indem es zunächst die Ergebnisse zusammenfasst (Abschnitt 10.1). Darauf erfolgt in Abschnitt 10.2 eine Reflektion in Hinblick auf die zugrundeliegende Forschungsmethodik. Der abschließende Ausblick (Abschnitt 10.3) skizziert das Potenzial für weitere Forschung, Erweiterungen sowie auch Einsatzgebiete des PERCOP-Frameworks.

### 10.1 Zusammenfassung

Das zentrale Ziel der vorliegenden Arbeit bestand in der Entwicklung eines generischen Frameworks, welches die Erstellung von Frameworks spezifiziert, die eine Echtzeit-informationsversorgung für die Verwendung softwareintensiver Systeme realisieren. Erreicht wurde die Zielsetzung zum einen durch die Entwicklung des intendierten Frameworks, zum anderen „allgemein“ durch die Bearbeitung der sich ergebenden Forschungsfragen.

Nachdem das erste Kapitel die Arbeit motiviert und das Ziel der Arbeit konkretisiert hat, wurden in Kapitel 2 der thematische Bezugsrahmen und damit die Grundlagen erarbeitet. Besondere Betrachtung galt dem Informationsmanagement, der Nutzung von Kontextinformationen sowie verwandten Arbeiten und Projekten.

Kapitel 3 präsentierte die Grundstruktur des intendierten Frameworks. Das Kapitel hat dazu zunächst die Bestandteile und anschließend die Architektur des Frameworks vorgestellt; es bildet damit die Basis für die nachfolgenden Kapitel und Überlegungen. Die wesentlichen Betrachtungsgegenstände in dem Kapitel waren die Benutzer und deren Informationsbedarf, Umgebungen und deren bereitgestellte Funktionalität sowie die Crowd und schließlich Empfehlungs- und Regelsysteme. Insbesondere die Trennung der einzelnen Teilaspekte des Konzepts stand im Fokus der Betrachtung und wurde in Hinblick auf die Beteiligungsmöglichkeiten strukturiert.

Das vierte Kapitel hat den Bedarf, die Beschaffung und die Bereitstellung von Informationen unter besonderer Betrachtung des Frameworks analysiert. Dazu wurden insbesondere die Konzepte des Information Retrieval untersucht und im Rahmen des Benutzerverhaltens hin auf das Framework angewendet.

Als elementarer Bestandteil des Konzepts wurde in Kapitel 5 das Potenzial der Masse untersucht. Dazu wurden zunächst die unterschiedlichen Felder betrachtet, in denen die

Beteiligung der Masse eine wichtige Rolle spielt. Dazu zählen kollektive Intelligenz, Crowdsourcing, interaktive Wertschöpfung und Open Source.

Anschließend wurden im Rahmen einer Motivationsanalyse die Gründe für die Beteiligung untersucht. Die Ergebnisse der Analyse wurden auf die Beteiligungsschnittstellen des Frameworks übertragen.

Die zentralen Ergebnisse des Kapitels beziehen sich somit zum einen auf die Zuordnung des Beteiligungspotenzials der Masse zu den Teilaspekten des Frameworks. Zum anderen wurden den unterschiedlichen Beteiligungsmöglichkeiten Motivationspotenziale zugeordnet.

Gegenstand des sechsten Kapitels war die Entwicklung eines Empfehlungssystems für das intendierte Framework. Zunächst wurden unterschiedliche Empfehlungsverfahren und -algorithmen analysiert. Anschließend wurde ein Empfehlungssystem konzipiert, welches eine Integration in das Framework unter Erfüllung der definierten Anforderungen ermöglicht. Wesentlicher Bestandteil sind Empfehlungsmodule und -Konfigurationen, die im Rahmen von PERCOP ein „Empfehlungssystem für Empfehlungssysteme“ repräsentieren.

Die Erzeugung und insbesondere die Quellen der Informationsartefakte bildeten den Gegenstand des siebten Kapitels. Zunächst wurden relevante Informationsquellen untersucht und darauf aufbauend Verfahren zur Ergänzung von semantischen Informationen betrachtet. In Hinblick auf die Nutzung innerhalb von PERCOP wurden sowohl eine Architektur als auch ein Prozess zur semantischen Anreicherung von Informationsartefakten entworfen.

Ziele des achten Kapitels waren der Entwurf und die Implementierung eines Prototyps. Die prototypische Umsetzung diente zum einen als Proof of concept und hat gezeigt, dass PERCOP technisch realisierbar ist. Der Prototyp umfasste alle erforderlichen Bestandteile, insbesondere eine Web-Plattform zur Integration der Funktionalität und der Crowd, einen Desktop-Client sowie verschiedene Sensor- und Recommender-Module.

Mit Hilfe des Prototyps konnten unterschiedliche Szenarios umgesetzt werden. Zur Validierung wurden im neunten Kapitel drei ausgewählte Fallbeispiele mit Hilfe des Prototyps realisiert. Die Fallbeispiele repräsentierten ganz bewusst unterschiedliche Domänen, Anwendungsfelder und Umgebungen. Die konkrete Abbildung von Anwendungsszenarios aus der Realwelt hat gezeigt, dass eine Umsetzung nicht nur auf technischer Ebene, sondern auch im Sinne der Nutzbarkeit realisiert werden kann. Dabei wurde insbesondere der „generische“ Aspekt des Frameworks validiert.

## 10.2 Reflexion aus der Perspektive der zugrundeliegenden Forschungsmethode

Ziel der gewählten Forschungsmethode war es, die Forschungsfragen der Arbeit zielgerichtet zu beantworten und damit die Zielsetzung der Arbeit zu erreichen. Die gestellten Ziele wurden auf diesem Wege erreicht; diese Einschätzung soll im Folgenden erläutert werden.

Zunächst wurde auf Basis von Studien und Diskussionen in Wissenschaft und Praxis die *Problemstellung* identifiziert und motiviert. Insbesondere die Ausführungen zu Aspekten des Informationsbedarfs und der -beschaffung bzw. zu den vorherrschenden Defiziten im Rahmen der Suchprozesse von Benutzern zeigen, dass die Problemstellung der Arbeit hoch relevant ist. Die Darstellung kontextbasierter Konzepte und der Nutzung kollektiver Intelligenz konkretisierte Ziel und Potenzial der zu entwickelnden Lösung.

*Ziel* der vorliegenden Arbeit war die Entwicklung eines (generischen) Frameworks zur Erstellung konkreter Frameworks, die eine kontextabhängige Informationsversorgung innerhalb softwareintensiver Systeme in Echtzeit realisieren können. Ziel der Frameworks ist somit die Förderung der Informationsversorgung zum einen in Hinblick auf die Bereitstellung und die Qualität der angebotenen Informationen. Zum anderen zielt das Konzept auf eine Reduzierung der Zeitdauer an, die in die manuelle Suche nach Informationen aktuell investiert wird. Um eine sowohl technologie- als auch domänenunabhängige Realisierung zu ermöglichen, wurde eine generische Lösung konzipiert und realisiert.

Die angewendete Forschungsmethode verlangt im Rahmen der Lösungserstellung den Entwurf und die Entwicklung von *Artefakten*. Diese wurden in der vorliegenden Arbeit durch ein Framework und ein Proof of concept in Gestalt eines Prototyps und seines Einsatzes in relevanten Szenarien realisiert. Die *Evaluation* und Validierung des erstellten Konzeptes erfolgte somit zum einen durch die Implementierung eines Prototyps, zum anderen durch Fallbeispiele, die unter Verwendung des Prototyps durchgeführt wurden. Die Inhalte bzw. Ausprägungen der Fallbeispiele wurden abgeleitet aus den zugrundeliegenden Forschungsfragen, ergänzt durch Feedback bei der Demonstration des Prototyps. Die durchgeführten Fallbeispiele belegen die grundsätzliche Fähigkeit des Frameworks, Anforderungen der Realwelt mit Hilfe einer konkreten PERCOP-Implementierung zu erfüllen.

Die Verbreitung der Forschungsaktivitäten erfolgt zum einen durch Veröffentlichung der Arbeit und des Prototyps. Zum anderen wurden bereits Teilergebnisse auf Konferenzen



und Tagungen präsentiert und veröffentlicht (vgl. Beul und Eicker 2009 sowie Beul und Eicker 2012) und mit einem Best Paper Award ausgezeichnet.

### 10.3 Ausblick

Zahlreiche Möglichkeiten der Anwendbarkeit von PERCOP wurden im Rahmen der vorliegenden Arbeit untersucht. Der Fokus der Arbeit lag auf der Konzeption eines „generischen“ Frameworks zur Erstellung konkreter (domänenspezifischer) Frameworks in softwareintensiven Umgebungen. Dabei wurden insbesondere Anwendungssysteme und deren Nutzung durch Endanwender betrachtet. In zukünftigen Forschungsarbeiten können weitere Umgebungstypen identifiziert und in Hinblick auf die Anwendbarkeit von PERCOP untersucht werden.

Für PERCOP sind insbesondere Erweiterungsmöglichkeiten im Rahmen der Kontextmodellierung relevant. Eine intensive Untersuchung der Nutzung von Ontologien und anderen Technologien zur Kontextmodellierung in Verbindung mit konkreten Domänen wären sicherlich sinnvolle ergänzende Forschungstätigkeiten. Zudem könnte eine noch differenziertere Betrachtung von Informationsbedarfen und Information-Retrieval-Aktivitäten innerhalb spezifischer Domänen eventuell weitere Erkenntnisse liefern, die in das Framework bzw. in die darauf basierenden Implementierungen einfließen könnten.

Im Hinblick auf die Bewertungen der durch PERCOP-Umsetzungen konkret dem Benutzer präsentierten Informationen wurde im Rahmen dieser Arbeit eine „einfache“ manuelle (aktive) binäre Bewertung durch die Benutzer realisiert. Möglicherweise kann über Verweildauern oder die Dokumentenauswahl automatisiert (passiv) auf weitere Informationen zur Bewertung geschlossen werden.

Schließlich ist denkbar, das Framework in Richtung aktiver Systeme zu erweitern, bei denen Informationen nicht nur präsentiert werden, sondern basierend auf den Informationen bzw. dem aktuellen Benutzerkontext aktiv in die Handlung des Benutzers eingegriffen wird. Natürlich wären dazu intensive Studien erforderlich, um die Akzeptanz solcher Eingriffe zu validieren.

Die Nutzung des Frameworks innerhalb einer geschlossenen Umgebung - beispielsweise in einem Unternehmen oder in einer Softwareentwicklungsabteilung - bietet die Möglichkeit, bereits bestehende Softwaresysteme als Datenbasis für Informationsartefakte zu nutzen, z. B. durch die Integration eines (internen) Dokumentenmanagement-Systems oder einer Knowledge Base.

Erweiterungsmöglichkeiten des Prototyps ergeben sich auch bezüglich der Unterstützung unterschiedlicher Plattformen und bei der Umsetzung konkreter Frameworks. Auch die Konfiguration und Zusammenstellung domänenspezifischer Frameworks mit dem Ziel einer effizienten Installation stellt möglicherweise eine sinnvolle Ergänzung dar. Auch die Beteiligung der Crowd könnte noch erweitert werden, beispielsweise in Form von aktiven Bewertungsmöglichkeiten von Sensoren und Recommender-/Konfigurationen.

Im Rahmen der Evaluation sind im nächsten Schritt Experimente auf Basis der Ergebnisse des Prototyps und der Fallbeispiele denkbar, mit denen konkrete Aussagen zum Benutzerverhalten und zur Zeitersparnis getroffen werden können. Vor allem das Verhältnis zwischen der Komplexität des Frameworks auf der einen Seite und der Beteiligung durch die Crowd auf der anderen Seite könnte detaillierter analysiert werden.

Der Autor wird die Weiterentwicklung und Nutzung des PERCOP-Projekts über die dazu eingerichtete Website <http://www.personalcopilot.com> motivieren und verfolgen.

## Literaturverzeichnis

Aamodt, Agnar; Nygård, Mads (1995): Different roles and mutual dependencies of data, information, and knowledge - An AI perspective on their integration. In: *Data & Knowledge Engineering* 16 (3), S. 191–222. DOI: 10.1016/0169-023X(95)00017-M.

Abecker, Andreas; Bernardi, Ansgar; Hinkelmann, Knut; Kühn, Otto; Sintek, Michael (2000): Context-Aware, Proactive Delivery of Task-Specific Information: The KnowMore Project. In: *Information Systems Frontiers* 2 (3-4), S. 253–276. DOI: 10.1023/A:1026564510897.

Abowd, Gregory D.; Dey, Anind K. (1999): Towards a Better Understanding of Context and Context-Awareness. In: Proceedings of the 1st international symposium on Handheld and Ubiquitous Computing. London, UK: Springer-Verlag, S. 304–307.

Adida, Ben; Herman, Ivan; Sporny, Manu; Birbeck, Mark (2012): RDFa 1.1 Primer. Rich Structured Data Markup for Web Documents. Online verfügbar unter <http://www.w3.org/TR/xhtml-rdfa-primer/>, zuletzt geprüft am 09.08.2016.

Adomavicius, Gediminas; Sankaranarayanan, Ramesh; Sen, Shahana; Tuzhilin, Alexander (2005): Incorporating contextual information in recommender systems using a multidimensional approach. In: *ACM Transactions on Information Systems* 23 (1), S. 103–145.

Adomavicius, Gediminas; Tuzhilin, Alexander (2011): Context-Aware Recommender Systems. In: Francesco Ricci, Lior Rokach, Bracha Shapira und Paul B. Kantor (Hg.): *Recommender Systems Handbook*. Boston, MA: Springer US, S. 217–253.

Alby, Tom (2008): *Web 2.0. Konzepte, Anwendungen, Technologien*. 3. Aufl. München: Hanser.

Alderfer, Clayton P. (1969): An empirical test of a new theory of human needs. In: *Organizational Behavior and Human Performance* 4 (2), S. 142–175. DOI: 10.1016/0030-5073(69)90004-X.

Alderfer, Clayton P. (1972): *Existence, relatedness, and growth: human needs in organizational settings*: Free Press.

Alpar, Paul; Keßler, Steffen; Blaschke, Steffen (2007): *Web 2.0. Neue erfolgreiche Kommunikationsstrategien für kleine und mittlere Unternehmen*. Wiesbaden: Hess. Ministerium für Wirtschaft, Verkehr und Landesentwicklung (Hessen-Media, 57).

Anderson, Chris (2004): The Long Tail. Online verfügbar unter <http://www.wired.com/wired/archive/12.10/tail.html>, zuletzt geprüft am 09.08.2016.

Ardelt, Rudolf G.; Ehrenberg, Dieter; Heinrich, Lutz Jürgen (2011): *Geschichte der Wirtschaftsinformatik. Entstehung und Entwicklung einer Wissenschaftsdisziplin*. Berlin [u. a.]: Springer.

Atkinson, John W.; Birch, David (1978): *An introduction to motivation*. 2. Aufl. New York: Van Nostrand.

Bächle, Michael (2008): Ökonomische Perspektiven des Web 2.0 – Open Innovation, Social Commerce und Enterprise 2.0. In: *Wirtschaftsinformatik* 50 (2), S. 129–132. DOI: 10.1365/s11576-008-0024-2.

Back, Andrea; Gronau, Norbert; Tochtermann, Klaus (Hg.) (2009): Web 2.0 in der Unternehmenspraxis. Grundlagen, Fallstudien und Trends zum Einsatz von Social-Software. 2. Aufl. München: Oldenbourg.

Baldauf, Matthias; Dustdar, Schahram; Rosenberg, Florian (2007): A survey on context-aware systems. In: *International Journal of Ad Hoc and Ubiquitous Computing* 2 (4), S. 263–277. DOI: 10.1504/IJAHUC.2007.014070.

Barbutto, John E.; Scholl, Richard W. (1998): Motivation sources inventory: development and validation of new scales to measure an integrative taxonomy of motivation. In: *Psychological Reports* 82 (3), S. 1011–1022. DOI: 10.2466/pr0.1998.82.3.1011.

Bardram, Jakob E. (2005): The Java Context Awareness Framework (JCAF) – A Service Infrastructure and Programming Framework for Context-Aware Applications. In: David Hutchison, Takeo Kanade, Josef Kittler, Jon M. Kleinberg, Friedemann Mattern, John C. Mitchell et al. (Hg.): *Pervasive Computing*, Bd. 3468. Berlin, Heidelberg: Springer (Lecture notes in computer science), S. 98–115.

Bartels, Patrick; Pomes, Robert (2007): Informationsmanagement und -sicherheit. Unterlagen zur Vorlesung. Universität Hannover. Online verfügbar unter [http://archiv.iwi.uni-hannover.de/cms/images/stories/upload/lv/sosem07/IM-IS/1\\_im\\_002.pdf](http://archiv.iwi.uni-hannover.de/cms/images/stories/upload/lv/sosem07/IM-IS/1_im_002.pdf), zuletzt geprüft am 09.08.2016.

Bates, Marcia J. (1989): The design of browsing and berrypicking techniques for the online search interface. In: *Online Information Review* 13 (5), S. 407–424. DOI: 10.1108/eb024320.

Baumöl, Ulrike (2014a): Enzyklopädie der Wirtschaftsinformatik. Aufgaben des Informationsmanagements. Hg. v. Karl Kurber, Jörg Becker, Norbert Gronau, Elmar Sinz und Leena Suhl. Oldenbourg Wissenschaftsverlag GmbH. Online verfügbar unter <http://www.enzyklopaedie-der-wirtschaftsinformatik.de/lexikon/daten-wissen/Informationsmanagement/Informationsmanagement--Aufgaben-des/index.html/?searchterm=aufgaben%20des%20informationsmanagements>, zuletzt aktualisiert am 13.10.2014, zuletzt geprüft am 09.08.2016.

Baumöl, Ulrike (2014b): Enzyklopädie der Wirtschaftsinformatik. Informationsbedarf. Hg. v. Karl Kurber, Jörg Becker, Norbert Gronau, Elmar Sinz und Leena Suhl. Oldenbourg Wissenschaftsverlag GmbH. Online verfügbar unter <http://www.enzyklopaedie-der-wirtschaftsinformatik.de/wi-enzyklopaedie/lexikon/daten-wissen/Grundlagen-der-Informationsversorgung/Informationsbedarf/index.html/?searchterm=informationsbedarf>, zuletzt aktualisiert am 13.10.2014, zuletzt geprüft am 09.08.2016.

Berners-Lee, T.; Hendler, J.; Lassila, O. (2001): The Semantic Web: a new form of Web content that is meaningful to computers will unleash a revolution of new possibilities. In: *Scientific American*.

Beul, Michael; Eicker, Stefan (2009): "Don't Call Us, We Call You" - A Community Driven Approach for (Domain Independent) Context Driven Information Retrieval (CoDIR). In: Kokou Yetongnon, Richard Chbeir und Albert Dipanda (Hg.): The Fifth International Conference on Signal Image Technology & Internet Based Systems - SITIS 2009. Marrakesh, Morocco: IEEE Computer Society, S. 458–464. Online verfügbar unter <http://dx.doi.org/10.1109/SITIS.2009.77>.

Beul, Michael; Eicker, Stefan (2012): Crowdsourcing Supported Context Detection for Improving Information Search Activities. In: InfoWare 2012. International Multi-Conference on Computing in the Global Information Technology. Venice, Italy, June 24–29: IARIA, S. 254–259.

Bibliographisches Institut GmbH (Hg.): Stichwort Kontext. Duden online. Online verfügbar unter <http://www.duden.de/node/651291/revisions/1204503/view>, zuletzt geprüft am 09.08.2016.

Bibliographisches Institut GmbH (Hg.): Stichwort Situation. Duden online. Online verfügbar unter <http://www.duden.de/node/727052/revisions/1160483/view>, zuletzt geprüft am 09.08.2016.

Bobadilla, J.; Ortega, F.; Hernando, A.; Gutiérrez, A. (2013): Recommender systems survey. In: *Knowledge-Based Systems* 46, S. 109–132. DOI: 10.1016/j.knosys.2013.03.012.

Bonaccorsi, Andrea; Rossi, Cristina (2003): Altruistic Individuals, Selfish Firms? The Structure of Motivation in Open Source Software. In: *SSRN Journal*. DOI: 10.2139/ssrn.433620.

Borst, W. N. (1997): Construction of Engineering Ontologies for Knowledge Sharing and Reuse. Doctoral Thesis. University of Twente, Enschede, The Netherlands.

Boumans, J. M.; Bruck, Peter A. (2008): High performance multimedia. A reader on the technological, cultural and economic dynamics of multimedia. Tokyo, Amsterdam, Washington DC: Ohmsha; IOS Press.

Brabham, Daren C. (2010): MOVING THE CROWD AT THREADLESS. In: *Information, Communication & Society* 13 (8), S. 1122–1145. DOI: 10.1080/13691181003624090.

Brabham, Daren C. (2012): Motivations for Participation in a Crowdsourcing Application to Improve Public Engagement in Transit Planning. In: *Journal of Applied Communication Research* 40 (3), S. 307–328.

Brügge, Bernd (2004): Open-Source-Software. Eine ökonomische und technische Analyse. Berlin: Springer.

Budzik, Jay; Hammond, Kristian J. (2000): User interactions with everyday applications as context for just-in-time information access. In: IUI '00 Proceedings of the 5th international conference on Intelligent user interfaces. New Orleans, Louisiana, United States. New York, NY, USA: ACM, S. 44–51.

Burke, Robin (2002): Hybrid Recommender Systems. Survey and Experiments. In: *User Modeling and User-Adapted Interaction* 12 (4), S. 331–370.

Burke, Robin (2007): Hybrid web recommender systems. In: Peter Brusilovsky, Alfred Kobsa und Wolfgang Nejdl (Hg.): *The adaptive Web. Methods and strategies of Web personalization*. Berlin: Springer (State-of-the-art survey, 4321), S. 377–408.

Busse, S.; Kutsche, R. D. (2000): Modellierung informationslogistischer Anwendungen (Forschungsberichte des Fachbereichs Informatik, 2000-9).

Byström, Katriina; Järvelin, Kalervo (1995): Task complexity affects information seeking and use. In: *Information Processing & Management* 31 (2), S. 191–213. DOI: 10.1016/0306-4573(95)80035-R.

Celma, Òscar (2010): Music recommendation and discovery. The long tail, long fail, and long play in the digital music space. Berlin, Heidelberg: Springer.

Chen, Guanrong; Pham, Trung Tat (2001): Introduction to fuzzy sets, fuzzy logic, and fuzzy control systems. Boca Raton, FL: CRC Press. Online verfügbar unter <http://site.ebrary.com/lib/alltitles/docDetail.action?docID=10228596>.

Chen, H.; Perich, F.; Finin, T.; Joshi, A. (2004): SOUPA: standard ontology for ubiquitous and pervasive applications. In: *Proceedings of MobiQuitous 2004. The First Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services*. Boston, Massachusetts, USA, August 22 - 26. Piscataway, NJ: IEEE Service Center, S. 258–267.

Chen, Harry (2004): An Intelligent Broker Architecture for Pervasive Context-Aware Systems. PhD thesis. University of Maryland, Baltimore County.

Christian Fischer (2010): Auf dem Weg zu Kriterien zur Auswahl einer geeigneten Evaluationsmethode für Artefakte der gestaltungsorientierten Wirtschaftsinformatik. In: Stefan Klink, Agnes Koschmider, Marco von Mevius und Andreas Oberweis (Hg.): *EMISA 2010 - Einflussfaktoren auf die Entwicklung flexibler, integrierter Informationssysteme. Beiträge des Workshops der GI-Fachgruppe Entwicklungsmethoden für Informationssysteme und deren Anwendung (EMISA), 07.-08. 10. 2010 in Karlsruhe, Germ, Bd. 172: GI (LNI), S. 101–115*.

Chu, Heting (2003): Information representation and retrieval in the digital age. Medford, N.J.: Information Today (ASIST monograph series).

Collins, Allan M.; Loftus, Elizabeth F. (1975): A spreading-activation theory of semantic processing. In: *Psychological Review* 82 (6), S. 407–428. Online verfügbar unter <http://psycnet.apa.org/index.cfm?fa=search.displayRecord&uid=1976-03421-001>.

Czehak, Marcel (2010): Metadaten-Management zur Anreicherung webbasierter Daten um kontextspezifische Informationen. Diplomarbeit. Universität Duisburg-Essen, Essen.

Deci, Edward L.; Ryan, Richard M. (1985): Intrinsic motivation and self-determination in human behavior. 5. Aufl. New York: Plenum Press (Perspectives in social psychology).

Deiters, W.; Löffeler, T.; Pfennigschmidt, S. (2003): The information logistics approach toward user demand-driven information supply. In: Diomidis Spinellis (Hg.): *Cross-media service delivery. Conference on Cross Media Service Delivery*. Boston, Mass: Kluwer Acad.

Publ (The Kluwer international series in engineering and computer science, 740), S. 37–48.

Delphi Group (2002): Perspectives on Information Retrieval. Boston, MA.

Delphi Group (2004): Information Intelligence: Content Classification and the Enterprise Taxonomy Practice. Boston, MA.

Dey, Anind K. (2001): Understanding and Using Context. In: *Personal and Ubiquitous Computing* 5 (1), S. 4–7.

Doan, Anhui; Ramakrishnan, Raghu; Halevy, Alon Y. (2011): Crowdsourcing systems on the World-Wide Web. In: *Communications of the ACM* 54 (4), S. 86. DOI: 10.1145/1924421.1924442.

Dragunov, Anton N.; Dietterich, Thomas G.; Johnsrude, Kevin; McLaughlin, Matthew; Li, Lida; Herlocker, Jonathan L. (2005): TaskTracer: A desktop environment to support multi-tasking knowledge workers. In: Proceedings of the 10th international conference on Intelligent user interfaces. San Diego, California, USA, S. 75–82.

Erlhofer, Sebastian (2012): Suchmaschinen-Optimierung. Das umfassende Handbuch: Aktuell zu Google Panda und Penguin. 6. Aufl. Bonn: Galileo Press.

Felfernig, A.; Friedrich, G.; Schmidt-Thieme, L. (2007): Introduction to the IEEE Intelligent Systems. Special Issue: Recommender Systems (22(3)), S. 18–21.

Fensel, Dieter (2004): Ontologies: a silver bullet for knowledge management and electronic commerce. 2. Aufl. Berlin: Springer.

Ferstl, Otto K.; Sinz, Elmar J. (2008): Grundlagen der Wirtschaftsinformatik. 6. Aufl. München, Wien: Oldenbourg.

Frank, Ulrich (1998): Die Evaluation von Artefakten: Eine zentrale Herausforderung der Wirtschaftsinformatik. In: Tagungsband des Workshops "Evaluation und Evaluationsforschung in der Wirtschaftsinformatik: Johannes Kepler Universität Linz. Online verfügbar unter <http://www.wi-inf.uni-due.de/FGFrank/documents/Zeitschriftenartikel/LinzEvaluation.pdf>.

Frank, Ulrich (2012): Enzyklopädie der Wirtschaftsinformatik. Konstruktionsorientierter Forschungsansatz. Hg. v. Karl Kurber, Jörg Becker, Norbert Gronau, Elmar Sinz und Leena Suhl. Oldenbourg Wissenschaftsverlag GmbH. Online verfügbar unter <http://enzyklopaedie-der-wirtschaftsinformatik.de/wi-enzyklopaedie/lexikon/uebergreifendes/Forschung-in-WI/Konstruktionsorientierter-Forschungsansatz>, zuletzt aktualisiert am 10.10.2012, zuletzt geprüft am 09.08.2016.

Fries, Rüdiger (2007): Suchverhalten im Internet. Studie über Suchstrategien im Web. Saarbrücken: VDM Verlag Dr. Müller. Online verfügbar unter [http://deposit.d-nb.de/cgi-bin/dokserv?id=2918455&prov=M&dok\\_var=1&dok\\_ext=htm](http://deposit.d-nb.de/cgi-bin/dokserv?id=2918455&prov=M&dok_var=1&dok_ext=htm).

Frommholz, I.; Fuhr, N. (2006): Probabilistic, object-oriented logics for annotation-based retrieval in digital libraries. In: Proceedings of the 6th ACM/IEEE-CS Joint Conference on Digital Libraries. Chapel Hill, NC, USA, June 11 - 15. New York, NY: Association for Computing Machinery, S. 55–64.

Fuhr, N. (1992): Konzepte zur Gestaltung zukünftiger Information-Retrieval-Systeme. In: Experimentelles und praktisches Information Retrieval: Universitätsverlag Konstanz, S. 59–75.

Fuhr, Norbert (2010): Einführung in Information Retrieval. Skriptum zur Vorlesung. Universität Duisburg-Essen. Online verfügbar unter [http://www.is.inf.uni-due.de/courses/ir\\_ss10/folien/skript\\_1-5.pdf](http://www.is.inf.uni-due.de/courses/ir_ss10/folien/skript_1-5.pdf), zuletzt geprüft am 09.08.2016.

Gabler Verlag (Hg.) (2016): Gabler Wirtschaftslexikon, Stichwort: Metawissen. Online verfügbar unter <http://wirtschaftslexikon.gabler.de/Archiv/78592/metawissen-v6.html>, zuletzt geprüft am 09.08.2016.

Gabriel, Roland (2013a): Enzyklopädie der Wirtschaftsinformatik. Operatives Informationssystem. Hg. v. Karl Kurber, Jörg Becker, Norbert Gronau, Elmar Sinz und Leena Suhl. Oldenbourg Wissenschaftsverlag GmbH. Online verfügbar unter <http://www.enzyklopaedie-der-wirtschaftsinformatik.de/wi-enzyklopaedie/lexikon/uebergreifendes/Kontext-und-Grundlagen/Informationssystem/operatives-informationssystem/operatives-informationssystem>, zuletzt aktualisiert am 30.09.2013, zuletzt geprüft am 09.08.2016.

Gabriel, Roland (2013b): Enzyklopädie der Wirtschaftsinformatik. Strategisches Informationssystem. Hg. v. Karl Kurber, Jörg Becker, Norbert Gronau, Elmar Sinz und Leena Suhl. Oldenbourg Wissenschaftsverlag GmbH. Online verfügbar unter <http://www.enzyklopaedie-der-wirtschaftsinformatik.de/wi-enzyklopaedie/lexikon/uebergreifendes/Kontext-und-Grundlagen/Informationssystem/Strategisches-Informationssystem>, zuletzt aktualisiert am 30.09.2013, zuletzt geprüft am 09.08.2016.

Gabriel, Roland (2013c): Enzyklopädie der Wirtschaftsinformatik. Anwendungssystem. Hg. v. Karl Kurber, Jörg Becker, Norbert Gronau, Elmar Sinz und Leena Suhl. Oldenbourg Wissenschaftsverlag GmbH. Online verfügbar unter <http://www.enzyklopaedie-der-wirtschaftsinformatik.de/wi-enzyklopaedie/lexikon/uebergreifendes/Kontext-und-Grundlagen/Anwendungssystem>, zuletzt aktualisiert am 16.10.2013, zuletzt geprüft am 09.08.2016.

Gabriel, Roland (2013d): Enzyklopädie der Wirtschaftsinformatik. Informationssystem. Hg. v. Karl Kurber, Jörg Becker, Norbert Gronau, Elmar Sinz und Leena Suhl. Oldenbourg Wissenschaftsverlag GmbH. Online verfügbar unter <http://www.enzyklopaedie-der-wirtschaftsinformatik.de/wi-enzyklopaedie/lexikon/uebergreifendes/Kontext-und-Grundlagen/Informationssystem>, zuletzt aktualisiert am 16.10.2013, zuletzt geprüft am 09.08.2016.

Gassmann, Oliver; Enkel, Ellen (2006): Open Innovation. Die Öffnung des Innovationsprozesses erhöht das Innovationspotenzial. In: *Zeitschrift Führung und Organisation (zfo)* (3), S. 132–138.

Gebert, Diether; Rosenstiel, Lutz von (2002): Organisationspsychologie. Person und Organisation. 5. Aufl. Stuttgart: Kohlhammer (Kohlhammer-Standards Psychologie).



Goldberg, David; Nichols, David; Oki, Brian M.; Terry, Douglas (1992): Using collaborative filtering to weave an information tapestry. In: *Communications of the ACM - Special issue on information filtering* 35 (12), S. 61–70. DOI: 10.1145/138859.138867.

Good, Nathaniel; Schafer, J. Ben; Konstan, Joseph A.; Borchers, Al; Sarwar, Badrul; Herlocker, Jon; Riedl, John (1999): Combining collaborative filtering with personal agents for better recommendations. In: Proceedings of the sixteenth national conference on Artificial intelligence and the eleventh Innovative applications of artificial intelligence conference innovative applications of artificial intelligence. Menlo Park, CA, USA: American Association for Artificial Intelligence (AAAI '99/IAAI '99), S. 439–446.

Griesbaum, Joachim (2007): Entwicklungstrends im Web Information Retrieval: Neue Potentiale für die Webrecherche durch Personalisierung & Web 2.0-Technologien. In: DGI (Hg.): 29. Online-Tagung der DGI, Frankfurt a.M. 10.-12. Oktober 2007.

Grob, Heinz Lothar; Reepmeyer, Jan-Armin; Bensberg, Frank (2004): Einführung in die Wirtschaftsinformatik. 5. Aufl. München: Vahlen.

Gruber, Thomas R. (1993): A translation approach to portable ontologies. In: *Knowledge Acquisition - Special issue: Current issues in knowledge modeling* 5 (2), S. 199–220.

Guarino, Nicola (1998): Formal Ontology and Information Systems. In: Proceedings of the First International Conference (FIOS'98). Formal Ontology in Information Systems, June 6-8. Trento, Italy: IOS Press, S. 3–15.

Gudehus, Timm (2011): Logistik. Grundlagen, Strategien, Anwendungen. 4. Aufl.: Springer-Verlag.

Güldenberger, Stefan (1997): Wissensmanagement und Wissenscontrolling in lernenden Organisationen. Ein systemtheoretischer Ansatz. 2. Aufl. Wiesbaden: Deutscher Universitäts-Verlag.

Hackman, J. Richard; Oldham, Greg R. (1980): Work redesign. Reading, MA: Addison-Wesley (Addison-Wesley series on organization development).

Hansen, Hans Robert; Neumann, Gustaf (2001): Grundlagen betrieblicher Informationsverarbeitung. 8. Aufl. Stuttgart: Lucius & Lucius (802).

Hars, Alexander; Ou, Shaosong (2002): Working for Free? Motivations for Participating in Open-Source Projects. In: *International Journal of Electronic Commerce* 6 (3), S. 25–39. Online verfügbar unter <http://dl.acm.org/citation.cfm?id=1286960.1286963>.

Hauptmann, Peter (1991): Sensoren. Prinzipien und Anwendungen. München: Hanser.

Hauschildt, Jürgen (1990): Methodische Anforderungen an die Ermittlung der Wissensbasis von Expertensystemen. In: *Die Betriebswirtschaft (DBW)* (4), S. 525–537.

Heinrich, Lutz J. (2005): Forschungsmethodik einer Integrationsdisziplin: Ein Beitrag zur Geschichte der Wirtschaftsinformatik. In: *NTM International Journal of History and Ethics of Natural Sciences, Technology and Medicine* 13 (2), S. 104–117. DOI: 10.1007/s00048-005-0211-9.

Heinrich, Lutz J.; Stelzer, Dirk (2011): Informationsmanagement. Grundlagen, Aufgaben, Methoden. München: Oldenbourg (Wirtschaftsinformatik).

Hemmje, Matthias (1999): Unterstützung von Information-Retrieval-Dialogen mit Informationssystemen durch interaktive Informationsvisualisierung. Dissertation. TU Darmstadt, Darmstadt. Online verfügbar unter <http://elib.tu-darmstadt.de/diss/000075>.

Henning, Michi (2008): The rise and fall of CORBA. In: *Commun. ACM* 51 (8), S. 52. DOI: 10.1145/1378704.1378718.

Hentze, Joachim (2005): Personalführungslehre. Grundlagen, Funktionen und Modelle der Führung. 4. Aufl. Stuttgart: UTB (UTB für Wissenschaft, 1374). Online verfügbar unter <http://www.utb-studi-e-book.de/9783838513744>.

Hentze, Joachim; Heinecke, Albert; Kammel, Andreas (2001): Allgemeine Betriebswirtschaftslehre aus Sicht des Managements. Stuttgart: UTB (2040).

Henzinger, Monika (2008): Tutorial: Web Information Retrieval. Online verfügbar unter [http://delab.csd.auth.gr/courses/\\_c\\_ir/webIRtutorial.pdf](http://delab.csd.auth.gr/courses/_c_ir/webIRtutorial.pdf), zuletzt geprüft am 09.08.2016.

Herlocker, Jonathan L.; Konstan, Joseph A.; Terveen, Loren G.; Riedl, John T. (2004): Evaluating collaborative filtering recommender systems. In: *ACM Transactions on Information Systems* 22 (1), S. 5–53.

Hertel, Guido; Niedner, Sven; Herrmann, Stefanie (2003): Motivation of software developers in Open Source projects: an Internet-based survey of contributors to the Linux kernel. In: *Research Policy* 32, S. 1159–1177.

Herzberg, Frederick; Mausner, Bernard; Snyderman, Barbara B. (1959): The motivation to work. 2. Aufl. New York, London: Wiley; Chapman & Hall.

Hevner, Alan R.; March, Salvatore T.; Park, Jinsoo; Ram, Sudha (2004): Design Science in Information Systems Research. In: *MIS Quarterly* 28 (1), p 75-105. Online verfügbar unter <http://www.jstor.org/stable/25148625>.

Hill, Will; Stead, Larry; Rosenstein, Mark; Furnas, George (1995): Recommending and evaluating choices in a virtual community of use. In: CHI '95 Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. Denver, Colorado, United States. New York, NY, USA: ACM Press/Addison-Wesley Publishing Co., S. 194–201.

Hippner, Hajo (2006): Bedeutung, Anwendung und Einsatzpotenziale von Social Software. In: K. Hildebrand und J. Hofmann (Hg.): Social Software, Bd. 43. Heidelberg: dpunkt.verlag (HMD 252), S. 6–16.

Hippner, Hajo; Wilde, Thomas (2005): Social Software. In: *Wirtschaftsinformatik* 47 (6), S. 441–444.

Höchstötter, Nadine (2007): Untersuchung des Suchverhaltens im Web. Interaktion von Internetnutzern mit Suchmaschinen. Hamburg: Kovač (Schriftenreihe Studien zum Konsumentenverhalten, 9).

Hölscher, Christoph; Strube, Gerhard (2000): Web search behavior of Internet experts and newbies. In: *Computer Networks* 33 (1-6), S. 337–346. DOI: 10.1016/S1389-1286(00)00031-1.

Holtz, Shel; Demopoulos, Ted (2006): *Blogging for business. Everything you need to know and why you should care.* Chicago, IL: Kaplan Pub.

Howe, Jeff (2006): *Crowdsourcing: A definition.* Online verfügbar unter [http://crowdsourcing.typepad.com/cs/2006/06/crowdsourcing\\_a.html](http://crowdsourcing.typepad.com/cs/2006/06/crowdsourcing_a.html), zuletzt aktualisiert am 02.06.2006, zuletzt geprüft am 09.08.2016.

Howe, Jeff (2008): *Crowdsourcing. How the power of the crowd is driving the future of business.* London: RH Business Books.

Hu, Yifan; Koren, Yehuda; Volinsky, Chris (2008): Collaborative filtering for implicit feedback datasets. In: *IEEE International Conference on Data Mining (ICDM 2008)*, S. 263–272.

Hull, R; Neaves, P; Bedford-Roberts, J (1997): Towards situated computing. In: *Digest of Papers. First International Symposium on Wearable Computers.* Cambridge, MA, USA, 13.10 - 14.10., S. 146–153.

Hussein, Tim; Ziegler, Jürgen (2011): Situationsgerechtes Recommending. In: *Informatik Spektrum* 34 (2), S. 143–152.

IEEE Standards Association (2007): 1362-1998 - IEEE Guide for Information Technology - System Definition - Concept of Operations (ConOps) Document.

Indulska, Jadwiga; Sutton, Peter (2003): Location management in pervasive systems. In: Chris Johnson (Hg.): *ACSW frontiers 2003. Proceedings of the Australasian Information Security Workshop and the Workshop on Wearable, Invisible, Context-Aware, Ambient, Pervasive and Ubiquitous Computing*, Adelaide, Australia, February 2003. Sydney: Australian Computer Society (21).

J. Stacy Adams (1965): Inequity In Social Exchange. In: Leonard Berkowitz (Hg.): *Advances in Experimental Social Psychology*, Bd. 2: Academic Press (*Advances in Experimental Social Psychology*), S. 267–299. Online verfügbar unter <http://www.sciencedirect.com/science/article/pii/S0065260108601082>.

Jannach, Dietmar (2011): *Recommender systems. An introduction.* New York, NY: Cambridge Univ. Press.

Jentzsch, Ric; Prekop, Paul (2001): A Conceptual Model of Collaborative Information Seeking. In: Don White (Hg.): *Knowledge Mapping and Management*: IGI Global, S. 98–113.

Jones, William P.; Teevan, Jaime (2007): *Personal information management.* Seattle: University of Washington Press.

Kaufmann, N.; Schulze, T.; Veit, D. (2011): More than fun and money. Worker Motivation in Crowdsourcing – A Study on Mechanical Turk. In: *Proceedings of the Seventeenth Americas Conference on Information Systems (AMCIS). The Seventeenth Americas Conference on Information Systems.* Detroit, Michigan, 04.08. - 07.08.

Klahold, André (2009): Empfehlungssysteme. Recommender Systems - Grundlagen, Konzepte und Lösungen. 1. Aufl. Wiesbaden: Vieweg+Teubner Verlag / GWV Fachverlage GmbH Wiesbaden.

Koch, Michael; Richter, Alexander (2007): Enterprise 2.0. Planung, Einführung und erfolgreicher Einsatz von Social Software in Unternehmen. München [u. a.]: Oldenbourg.

Koreimann, Dieter S. (1976): Methoden der Informationsbedarfsanalyse. Berlin: Gruyter, Walter de GmbH.

Kozinets, R. V.; Hemetsberger, A.; Schau, H. J. (2008): The Wisdom of Consumer Crowds: Collective Innovation in the Age of Networked Marketing. In: *Journal of Macromarketing* 28 (4), S. 339–354. DOI: 10.1177/0276146708325382.

Krcmar, Helmut (2010): Informationsmanagement. 5. Aufl. Berlin, Heidelberg: Springer.

Kurbel, Karl (2008): The making of information systems. Software engineering and management in a globalized world. Berlin: Springer.

Kurbel, Karl (2014): Enzyklopädie der Wirtschaftsinformatik. Software. Hg. v. Karl Kurber, Jörg Becker, Norbert Gronau, Elmar Sinz und Leena Suhl. Oldenbourg Wissenschaftsverlag GmbH. Online verfügbar unter <http://www.enzyklopaedie-der-wirtschaftsinformatik.de/wi-enzyklopaedie/lexikon/technologien-methoden/Software/software/?searchterm=systemnahe%20software>, zuletzt aktualisiert am 19.09.2014, zuletzt geprüft am 09.08.2016.

Kuropka, Dominik (2004): Modelle zur Repräsentation natürlichsprachlicher Dokumente. Ontologie-basiertes Information-Filtering-und-Retrieval mit relationalen Datenbanken. 1. Aufl. Berlin: Logos Verlag Berlin (Advances in information systems and management science, 10).

Lakhani, Karim; Wolf, Robert (2005): Why Hackers Do What They Do: Understanding Motivation and Effort in Free/Open Source Software Projects. In: Joesph Feller, Brian Fitzgerald, Scott Hissam und Karim Lakhani (Hg.): Perspectives on Free and Open Source Software. Cambridge: MIT Press.

Landwich, P.; Hemmje, M.; Fuhr, N. (2007): Ansatz zu einem konzeptionellen Modell für interaktive Information-Retrieval-Systeme mit Unterstützung von Informationsvisualisierung. In: Open Innovation. Proc. 10. Internationales Symposium für Informationswissenschaft (ISI), S. 327–332.

Lansdale, M. (1988): The psychology of personal information management. In: *Applied Ergonomics* 19 (1), S. 55–66.

Leimeister, Jan Marco (2010a): Collective Intelligence. In: *Business & Information Systems Engineering* (Volume 2, Number 4), S. 245–248. DOI: 10.1007/s12599-010-0114-8.

Leimeister, Jan Marco (2010b): Kollektive Intelligenz. In: *Wirtschaftsinformatik* (Vol. 2, Nr. 4), S. 239–242. DOI: 10.1007/s11576-010-0234-2.

Levashova, T.; Lundqvist, M.; Sandkuhl, K.; Smirnov, A. (2006): Context-based modelling of information demand: approaches from information logistics and decision support. In: Proceedings of the 14th European Conference on Information Systems.

Lévy, Pierre (1997): Collective intelligence. Mankind's emerging world in cyberspace. Cambridge, Mass: Perseus Books (Helix books). Online verfügbar unter <http://www.gbv.de/dms/bowker/toc/9780738202617.pdf>.

Lewandowski, Dirk (2005): Web Information Retrieval. Deutsche Gesellschaft für Informationswissenschaft und Informationspraxis.

Linoff, Gordon S.; Berry, Michael J. (2011): Data mining techniques. For marketing, sales, and customer relationship management. 3. Aufl. Indianapolis: Wiley Pub., Inc.

Lundqvist, Magnus (2007): Information demand and use. Improving information flow within small-scale business contexts. Linköping: Department of Computer and Information Science, Linköpings universitet.

Lundqvist, Magnus; Sandkuhl, Kurt (2004): Modelling Information Demand for Collaborative Engineering. In: Proceedings of 2nd Intl. Workshop on Challenges in Collaborative Engineering. School of Engineering, Jönköping University, JTH, Computer and Electrical Engineering; CenIT, S. 111–120.

Machill, Marcel (2003): Wegweiser im Netz. Qualität und Nutzung von Suchmaschinen. Gütersloh: Verl. Bertelsmann-Stiftung.

Malone, Thomas W.; Laubacher, Robert; Dellarocas, Chrysanthos (2009): Harnessing Crowds: Mapping the Genome of Collective Intelligence. In: *MIT Sloan Research Paper No. 4732-09*.

Mäntyjärvi, Jani (2003): Sensor-based context recognition for mobile applications. Ph.D. Thesis. University of Oulu, Espoo, Oulu.

Mäntyjärvi, Jani; Seppänen, Tapio (2002): Adapting Applications in Mobile Terminals Using Fuzzy Context Information. In: Fabio Paternò (Hg.): Lecture Notes in Computer Science. Berlin, Heidelberg: Springer Berlin Heidelberg, S. 95–107.

Martin, Nicole; Lessmann, Stefan; Voß, Stefan (2008): Crowdsourcing: Systematisierung praktischer Ausprägungen und verwandter Konzepte. In: Martin Bichler, Thomas Hess, Helmut Krcmar, Ulrike Lechner, Florian Matthes, Arnold Picot et al. (Hg.): Multikonferenz Wirtschaftsinformatik 2008. TU München in Garching, 26. - 28. Februar. Berlin: Gito-Verl.

Maslow, A. H. (1943): A theory of human motivation. In: *Psychological Review* 50, S. 370–396.

Maslow, Abraham H. (1971): The farther reaches of human nature. New York: Viking Press (Esalen book).

Maslow, Abraham Harold (1981): Motivation und Persönlichkeit. Reinbek b. Hamburg: Rowohlt-Taschenbuch-Verl. (Rororo Sachbuch, 7395).

Mattern, Friedemann (2003): Total vernetzt. Szenarien einer informatisierten Welt; 7. Berliner Kolloquium der Gottlieb Daimler- und Karl Benz-Stiftung. Berlin [u. a.]: Springer (Xpert.press).

Maurice, Florence (2009): Microformats. Semantik für Webseiten. München: Entwickler-Press.

McAfee, Andrew P. (2006): Enterprise 2.0: The Dawn of Emergent Collaboration. In: *MIT Sloan Management Review* 47 (3), S. 21–28. Online verfügbar unter <http://sloanreview.mit.edu/issue/spring-2006/>.

McClelland, D. C.; Atkinson, J. W.; Clark, R. A.; Lowell, E. L. (1953): The achievement motive. New York: Appleton-Century-Crofts.

microformats.org (2013): Introduction to Microformats. Online verfügbar unter <http://microformats.org/wiki/introduction>, zuletzt aktualisiert am 07.04.2013, zuletzt geprüft am 09.08.2016.

Middleton, Stuart E.; Shadbolt, Nigel R.; Roure, David C. de (2004): Ontological user profiling in recommender systems. In: *ACM Transactions on Information Systems* 22 (1), S. 54–88. DOI: 10.1145/963770.963773.

Mizzaro, Stefano (1997): Relevance: The whole history. In: *Journal of the American Society for Information Science* 48 (9), S. 810–832. DOI: 10.1002/(SICI)1097-4571(199709)48:9<810::AID-ASIS6>3.0.CO;2-U.

Mobasher, Bamshad; Burke, Robin; Bhaumik, Runa; Williams, Chad (2007): Toward trustworthy recommender systems: An analysis of attack models and algorithm robustness. In: *ACM Transactions on Internet Technology* 7 (4). DOI: 10.1145/1278366.1278372.

Murray, Henry Alexander (1938): Explorations in personality. Oxford: Oxford University Press.

Nguyen, An-Te; Denos, Nathalie; Berrut, Catherine (2007): Improving new user recommendations with rule-based induction on cold user data. In: Proceedings of the 2007 ACM conference on Recommender systems. New York, NY, USA: ACM, S. 121–128. Online verfügbar unter <http://doi.acm.org/10.1145/1297231.1297251>.

Nonaka, Ikujiro; Takeuchi, Hirotaka (1995): The knowledge-creating company. How Japanese companies create the dynamics of innovation. New York: Oxford University Press.

O'Reilly, Tim (2005): What Is Web 2.0. Design Patterns and Business Models for the Next Generation of Software. O'Reilly Verlag GmbH & Co.KG.

Oard, Douglas; Kim, Jinmook (1998): Implicit Feedback for Recommender Systems. In: in Proceedings of the AAAI Workshop on Recommender Systems, S. 81–83.

Oesterle, H.; Becker, J.; Frank, U.; Hess, T.; Karagiannis, D.; Krcmar, H. et al. (2010): Memorandum zur gestaltungsorientierten Wirtschaftsinformatik. In: *Zeitschrift für betriebswirtschaftliche Forschung* 6 (62), S. 664–672. Online verfügbar unter <http://www.alexandria.unisg.ch/publications/69397>.

Offermann, Philipp; Levina, Olga; Schönherr, Marten; Bub, Udo (2009): Outline of a design science research process. In: Proceedings of the 4th International Conference on Design Science Research in Information Systems and Technology. New York, NY, USA: ACM (DESRIST '09). Online verfügbar unter <http://doi.acm.org/10.1145/1555619.1555629>.

Open Source Initiative (OSI) (2007): The Open Source Definition. Online verfügbar unter <http://opensource.org/docs/osd>, zuletzt aktualisiert am 22.03.2007, zuletzt geprüft am 09.08.2016.

Ortner, Erich (2012): Enzyklopädie der Wirtschaftsinformatik. Geschichte der Wirtschaftsinformatik. Hg. v. Karl Kurber, Jörg Becker, Norbert Gronau, Elmar Sinz und Leena Suhl. Oldenbourg Wissenschaftsverlag GmbH. Online verfügbar unter <http://www.enzyklopaedie-der-wirtschaftsinformatik.de/wi-enzyklopaedie/lexikon/daten-wissen/Grundlagen-der-Informationsversorgung/Informationsbedarf/index.html/?searchterm=informationsbedarf>, zuletzt aktualisiert am 25.10.2012, zuletzt geprüft am 28.05.2013.

Palmisano, C.; Tuzhilin, A.; Gorgoglione, M. (2008): Using Context to Improve Predictive Modeling of Customers in Personalization Applications. In: *IEEE Transactions on Knowledge and Data Engineering* 20 (11), S. 1535–1549. DOI: 10.1109/TKDE.2008.110.

Papsdorf, Christian (2009): Wie Surfen zu Arbeit wird. Crowdsourcing im Web 2.0. Frankfurt am Main (Sozialwissenschaften 2009). Online verfügbar unter [http://www.content-select.com/index.php?id=bib\\_view&ean=9783593407500](http://www.content-select.com/index.php?id=bib_view&ean=9783593407500).

Park, Yoon-Joo; Tuzhilin, Alexander (2008): The long tail of recommender systems and how to leverage it. In: Proceedings of the 2008 ACM conference on Recommender systems - RecSys '08: ACM Press, S. 11–18.

Pascoe, Jason (1998): Adding Generic Contextual Capabilities to Wearable Computers. In: Proceedings of the 2nd IEEE International Symposium on Wearable Computers. Washington, DC, USA: IEEE Computer Society (ISWC '98). Online verfügbar unter <http://dl.acm.org/citation.cfm?id=857199.858020>.

Pazzani, Michael J. (1999): A Framework for Collaborative, Content-Based and Demographic Filtering. In: *Artificial Intelligence Review - Special issue on data mining on the Internet* 13 (5-6), S. 393–408. DOI: 10.1023/A:1006544522159.

Peerce, Jennifer; Shneiderman, Ben (2009): The Reader-to-Leader Framework: Motivating Technology-Mediated Social Participation. In: *AIS Transactions on Human-Computer Interaction* 1 (1), S. 13–32. Online verfügbar unter <http://aisel.aisnet.org/thci/vol1/iss1/5/>.

Peppers, Ken; Tuunanen, Tuure; Rothenberger, Marcus; Chatterjee, Samir (2007): A Design Science Research Methodology for Information Systems Research. In: *Journal of Management Information Systems* 24, S. 45–77. DOI: 10.2753/MIS0742-1222240302.

Picot, Arnold; Reichwald, Ralf; Wigand, Rolf T. (2003): Die grenzenlose Unternehmung. Information, Organisation und Management ; Lehrbuch zur Unternehmensführung im Informationszeitalter. 5. Aufl. Wiesbaden: Betriebswirtschaftlicher Verl. Gabler (Gabler-Lehrbuch).

Polanyi, Michael (1985): Implizites Wissen. 1. Aufl. Frankfurt am Main: Suhrkamp (Suhrkamp Taschenbuch Wissenschaft, 543).

Poslad, Stefan (2009): Ubiquitous computing. Smart devices, environments and interactions. 1. Aufl. Chichester: Wiley.

Probst, Gilbert J. B.; Raub, Steffen; Romhardt, Kai (1999): Wissen managen. Wie Unternehmen ihre wertvollste Ressource optimal nutzen. 3. Aufl. Wiesbaden: Gabler.

Ramezani, M.; Bergman, L.; Thompson, R.; Burke, R.; Mobasher, B. (2008): Selecting and Applying Recommendation Technology. In: Proceedings of International Workshop on Recommendation and Collaboration, in Conjunction with 2008 International ACM Conference on Intelligent User Interfaces (IUI 2008).

Reeve, Lawrence; Han, Hyoil (2005): Survey of semantic annotation platforms. In: Lorie M. Liebrock (Hg.): Proceedings of the 2005 ACM symposium on Applied computing. SAC '05 The 2005 ACM Symposium on Applied Computing. Santa Fe, New Mexico, March 13 - 17. New York, NY, USA: ACM, S. 1634–1638.

Reichwald, Ralf; Piller, Frank (2009): Interaktive Wertschöpfung. Open Innovation, Individualisierung und neue Formen der Arbeitsteilung. 2. Aufl. Wiesbaden: Gabler Verlag / GWV Fachverlage GmbH Wiesbaden. Online verfügbar unter <http://dx.doi.org/10.1007/978-3-8349-9440-0>.

Reichwald, Ralf; Piller, Frank Thomas (2006): Interaktive Wertschöpfung. Open Innovation, Individualisierung und neue Formen der Arbeitsteilung. 1. Aufl. Wiesbaden: Gabler.

Reif, Gerald (2006): Semantische Annotation. In: Tassilo Pellegrini und Andreas Blumauer (Hg.): Semantic Web. Berlin, Heidelberg: Springer Berlin Heidelberg (X.media.press), S. 405–418.

Renneberg, Volker (2010): Adaptives, baukastenbasiertes Recommendersystem. Univ. der Bundeswehr München, Lohmar, Neubiberg.

Resnick, Paul; Iacovou, Neophytos; Suchak, Mitesh; Bergstrom, Peter; Riedl, John (1994): GroupLens: an open architecture for collaborative filtering of netnews. In: CSCW '94 Proceedings of the 1994 ACM conference on Computer supported cooperative work. Chapel Hill, North Carolina, United States. New York, NY: ACM, S. 175–186.

Resnick, Paul; Varian, Hal R. (1997): Recommender systems. In: *Communications of the ACM* 40 (3), S. 56–58.

Riege, Christian; Saat, Jan; Bucher, Tobias (2009): Systematisierung von Evaluationsmethoden in der gestaltungsorientierten Wirtschaftsinformatik. In: Jörg Becker, Helmut Krcmar und Björn Niehaves (Hg.): Wissenschaftstheorie und gestaltungsorientierte Wirtschaftsinformatik. Heidelberg: Physica-Verlag HD, S. 69–86.

Ryan, N. S.; Pascoe, J.; Morse, D. R. (1998): Enhanced reality fieldwork: the context-aware archaeological assistant. In: V. Gaffney, M. van Leusen und S. Exxon (Hg.): Enhanced Reality Fieldwork: the Context-aware Archaeological Assistant. Oxford: Tempus Reparatum (British Archaeological Reports).



Ryan, Richard M.; Deci, Edward L. (2000): Intrinsic and Extrinsic Motivations: Classic Definitions and New Directions. In: *Contemporary Educational Psychology* 25 (1), S. 54–67. DOI: 10.1006/ceps.1999.1020.

Salton, Gerard; MacGill, Michael J. (1987): Information retrieval - Grundlegendes für Informationswissenschaftler. Hamburg: McGraw-Hill (McGraw-Hill-Texte).

Sandkuhl, Kurt (2009): Demand-oriented Supply of Digital Content. Tutorial at RCDL 2009. In: *Digital Libraries: Advanced methods and technologies, digital collections : Proceedings of the XI. All-Russian Research Conference*.

Scerri, Simon; Abela, Charlie; Montebello, Matthew (2005): semantExplorer: A Semantic Web Browser. In: P. Isaías und M.B. Nunes (Hg.): IADIS International Conference WWW/Internet 2005.

Schafer, J. Ben; Frankowski, Dan; Herlocker, Jon; Sen, Shilad (2007): Collaborative Filtering Recommender Systems. In: Peter Brusilovsky, Alfred Kobsa und Wolfgang Nejdl (Hg.): *The adaptive Web. Methods and strategies of Web personalization*, Bd. 4321. Berlin: Springer (State-of-the-art survey, 4321), S. 291–324.

Schafer, J. Ben; Konstan, Joseph; Riedl, John (1999): Recommender systems in e-commerce. In: *Proceedings of the 1st ACM conference on Electronic commerce - EC '99*: ACM Press, S. 158–166.

Schaumburg, Hanno (1992): Sensoren. Stuttgart: Teubner (Werkstoffe und Bauelemente der Elektrotechnik, 3).

Schein, Andrew I.; Popescul, Alexandrin; Ungar, Lyle H.; Pennock, David M. (2002): Methods and metrics for cold-start recommendations. In: *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*. Tampere, Finland. New York, NY, USA: ACM, S. 253–260.

Schiessle, Edmund (1992): Sensortechnik und Meßwertaufnahme. 1. Aufl. Würzburg: Vogel-Verl (Vogel-Fachbuch).

Schilit, B.; Adams, N.; Want, R. (1994): Context-Aware Computing Applications. In: *WMCSA '94 Proceedings of the 1994 First Workshop on Mobile Computing Systems and Applications*. Santa Cruz, California, USA, S. 85–90.

Schmid, Michael (2004): Rationales Handeln und soziale Prozesse. Beiträge zur soziologischen Theoriebildung. Wiesbaden: VS Verlag für Sozialwissenschaften.

Schmidt, Albrecht (2002): Ubiquitous Computing - Computing in Context. Ph.D. Thesis. Lancaster University, UK. Computing Department.

Schmidt, Albrecht; Aidoo, Kofi Asante; Takaluoma, Antti; Tuomela, Urpo; van Laerhoven, Kristof; van de Velde, Walter (1999): Advanced Interaction in Context. In: *Proceedings of the 1st international symposium on Handheld and Ubiquitous Computing*. London, UK: Springer-Verlag (HUC '99), S. 89–101. Online verfügbar unter <http://dl.acm.org/citation.cfm?id=647985.743849>.

Schwittek, Widura; Eicker, Stefan (2012): Decision Support for Off-the-Shelf Software Selection in Web Development Projects. In: Michael Grossniklaus und Manuel Wimmer

(Hg.): Current Trends in Web Engineering, Bd. 7703: Springer Berlin Heidelberg (Lecture notes in computer science), S. 238–243. Online verfügbar unter [http://dx.doi.org/10.1007/978-3-642-35623-0\\_26](http://dx.doi.org/10.1007/978-3-642-35623-0_26).

Shardanand, Upendra; Maes, Pattie (1995): Social information filtering: algorithms for automating “word of mouth”. In: CHI '95 Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. Denver, Colorado, United States. New York, NY, USA: ACM Press/Addison-Wesley Publishing Co., S. 210–217.

Simon, Herbert A. (1959): Theories of Decision-Making in Economics and Behavioral Science. In: American Economic Association (Hg.): The American Economic Review (49, 3), S. 253–283.

Spink, Amanda; Jansen, Bernard J. (2004): Web search. Public searching of the Web. Dordrecht, Netherlands, Boston: Kluwer Academic Publishers (Information science and knowledge management, 6).

Staehle, Wolfgang H.; Conrad, Peter; Sydow, Jörg (1999): Management. Eine verhaltenswissenschaftliche Perspektive. 8. Aufl. München: Vahlen (Vahlens Handbücher der Wirtschafts- und Sozialwissenschaften).

Stahlknecht, Peter; Hasenkamp, Ulrich (2002): Einführung in die Wirtschaftsinformatik. 10. Aufl. Berlin [u. a.]: Springer (Springer-Lehrbuch).

Stelzer, Dirk (2001): Informationsbedarf. In: Peter Mertens (Hg.): Lexikon der Wirtschaftsinformatik. 4. Aufl. Berlin - Heidelberg - New York: Springer, S. 238–239.

Stock, Wolfgang G. (2007): Information retrieval. Informationen suchen und finden. München: Oldenbourg (Einführung in die Informationswissenschaft, 1).

Strang, Thomas; Linnhoff-Popien, Claudia (2004): A Context Modeling Survey. In: Workshop on Advanced Context Modelling, Reasoning and Management, UbiComp 2004 - The Sixth International Conference on Ubiquitous Computing, Nottingham/England.

Strauch, Bernhard; Winter, Robert (2002): Vorgehensmodell für die Informationsbedarfsanalyse im Data Warehousing. In: Eitel Maur und Robert Winter (Hg.): Vom Data Warehouse zum Corporate Knowledge Center. Heidelberg: Physica-Verlag HD, S. 359–378.

Studer, Rudi; Benjamins, V. Richard; Fensel, Dieter (1998): Knowledge engineering: principles and methods. In: *Data & Knowledge Engineering* 25 (1-2), S. 161–197. DOI: 10.1016/S0169-023X(97)00056-6.

Surowiecki, James (2005): The wisdom of crowds. 1. Aufl. New York, NY: Anchor Books.

Tapscott, Don; Williams, Anthony D. (2007): Wikinomics. Die Revolution im Netz. München: Hanser.

Teevan, Jaime; Jones, William; Bederson, Benjamin B. (2006): Personal Information Management. In: *Communications of the ACM* 49 (1), S. 40. DOI: 10.1145/1107458.1107488.

Tintarev, Nava; Masthoff, Judith (2007): A Survey of Explanations in Recommender Systems. In: 2007 IEEE 23rd International Conference on Data Engineering Workshop: IEEE, S. 801–810.

Twitter, Inc (2013): Über Twitter. Online verfügbar unter <https://twitter.com/about>, zuletzt geprüft am 20.06.2013.

Valerio, Andrea; Succi, Giancarlo; Fenaroli, Massimo (1997): Domain analysis and framework-based software development. In: *SIGAPP Appl. Comput. Rev.* 5 (2), S. 4–15. DOI: 10.1145/297075.297081.

Voß, Stefan; Gutenschwager, Kai (2001): Informationsmanagement. Berlin: Springer.

Vroom, Victor H. (1964): Work and motivation. New York: Wiley.

Walster, Elaine; Walster, G. William; Berscheid, Ellen (1978): Equity. Theory and research. Boston: Allyn and Bacon.

Wang, X. H.; Zhang, D. Q.; Gu, T.; Gung, H. K. (2004): Ontology Based Context Modeling and Reasoning using OWL. In: Proceedings of the Second IEEE Annual Conference on Pervasive Computing and Communications workshops. Orlando, Florida, Mar. 14 - 17. Computer Society; University of Texas. Los Alamitos, Calif.: IEEE Computer Society, S. 18–22.

Weiner, Bernard (1994): Motivationspsychologie. 3. Aufl. Weinheim: Beltz, Psychologie-Verl.-Union.

Weiser, Mark (1991): The Computer for the 21st Century. In: *Scientific American* 265 (3), S. 94–104. DOI: 10.1038/scientificamerican0991-94.

Weiss, Andrew (1995): Human Capital vs. Signalling Explanations of Wages. In: *The Journal of Economic Perspectives* 9 (4), p 133-154. Online verfügbar unter <http://www.jstor.org/stable/2138394>.

Weissenberger-Eibl, Marion A.; Borchers, Carsten (2007): Wissen, verstehen, finden. In: *wissensmanagement* (2), S. 28–31.

Wersig, Gernot (1973): Zur Systematik der Benutzerforschung. In: *Nachrichten für Dokumentation* (1), S. 10–14.

Wilde, Thomas; Hess, Thomas (2007): Forschungsmethoden der Wirtschaftsinformatik - Eine empirische Untersuchung. In: *Wirtschaftsinformatik* 49 (4), S. 280–287.

Wilson, T. D. (1981): On User Studies and Information Needs. In: *Journal of Documentation* 37 (1), S. 3–15. Online verfügbar unter <http://www.eric.ed.gov/ERICWebPortal/detail?accno=EJ248909>.

Wolf, Petra (2007): Führungsinformationen für das Kommunalmanagement. 1. Aufl.: DUV Deutscher Universitäts-Verlag. Online verfügbar unter <http://www.gbv.ebib.com/patron/FullRecord.aspx?p=752287>.

Wörndl, Wolfgang; Schlichter, Johann (2012): Enzyklopädie der Wirtschaftsinformatik. Persönliches Informationsmanagement. Hg. v. Karl Kurber, Jörg Becker, Norbert Gronau, Elmar Sinz und Leena Suhl. Oldenbourg Wissenschaftsverlag GmbH. Online verfügbar

unter <http://www.encyklopaedie-der-wirtschaftsinformatik.de/lexikon/daten-wissen/Informationsmanagement/Informationsmanagement--Aufgaben-des/Personliches-Informationsmanagement>, zuletzt aktualisiert am 10.10.2012.

Zadeh, L.A (1965): Fuzzy sets. In: *Information and Control* 8 (3), S. 338–353. DOI: 10.1016/S0019-9958(65)90241-X.

Zerfaß, Ansgar; Boelter, Dietrich (2005): Die neuen Meinungsmacher. Weblogs als Herausforderung für Kampagnen, Marketing, PR und Medien. 1. Aufl. Graz: Nausner und Nausner (FastBook, 4).

Zerfaß, Ansgar; Bogosyan, Janine (2007): Blogstudie 2007. Informationssuche im Internet – Blogs als neues Recherchetool (Ergebnisbericht).

# Anhang

## A.1: Analyse der Beteiligung im Rahmen von PERCOP

PERCOP Beteiligung	Art der Kollaboration		Eigenschaften				Kernfragen		
	Explizit	Implizit	Adressiertes Problem	Ausprägung des CS-Systems	Tätigkeiten	Benutzer-Rolle	Wie können sich die einzelnen Benutzer beteiligen?	Wie werden die Beiträge der einzelnen Benutzer zur Gesamtlösung zusammengeführt?	Wie werden die Beiträge der Benutzer und deren Beträge evaluiert?
Umgebungen und Funktionalitäten definieren	X		Building a collection of items	Sharing Evaluating	Defining Environments and functionalities	content provider perspective provider	Web-Portal: Umgebungen und Funktionen definieren	Umgebungen und deren Funktionalitäten dienen den anderen Aktivitäten als Basis	Kontrolle durch CS-Systeme
Regeln erstellen	X	(X)	Building a collection of items	Sharing Evaluating	Creation, sharing and evaluation of rules	content provider perspective provider	Web-Portal: Regeln über Aspektentypen definieren, implizit über Aufzeichnung von Aktivitäten	Regeln verbinden, bestehende Elemente perspektivieren, Funktionalitäten definieren, implizit über Aspektentypen definieren, implizit über Aufzeichnung von Aktivitäten	Kontrolle durch "Umgebungs-Ersteller"
Funktionalitäten zuordnen	X		Building a collection of items	Sharing Evaluating	Creation, sharing and evaluation of mappings	content provider perspective provider	Web-Portal: Zuordnung von Aktionen und Informationen zu Funktionalitäten	Funktionalitäten werden direkt mit Informationsartefakten verknüpft. Qualitätskontrolle erfolgt aktiv über Benutzer	Kontrolle durch andere Benutzer
Informationen zuordnen	X		Building physical artifacts	Sharing Evaluating	Creation, sharing and evaluation of mappings	content provider perspective provider	Web-Portal: Zuordnung von Aktionen und Informationen zu Funktionalitäten	Informationen werden direkt mit Funktionalitäten verknüpft. Qualitätskontrolle erfolgt aktiv über Benutzer	Implizite Kontrolle durch Bewertungen
Informationsartefakte erstellen	X		Building physical artifacts	Sharing	Creation of information artifacts	content provider	Erstellen von Informationsartefakten innerhalb beliebiger (externer) Systeme	Bestehende Informationsartefakte werden durch Verknüpfung integriert	Kontrolle durch externe Systeme, bei denen die Artefakte liegen
Informationsartefakte semantisch anreichern	X	X	Evaluating (tag artifacts)	Evaluating (tag artifacts)	Manual/automatic tagging, Rating of "Answers"	all roles possible	Ergänzung von Metadaten zu bestehenden Artefakten	Semantische Anreicherung Situationsgerichtetes Auffinden von Artefakten	Kontrolle durch externe Systeme, bei denen die Artefakte liegen
Sensoren entwickeln	X		Building physical artifacts	Building physical artifacts	Software development	content provider	Entwicklung von Software-Komponenten in Form von Open-Source-Projekten	Die Sensoren übertragen ihre Daten an ein Informations-Broker und können dort weiterverwendet werden	Kontrolle durch andere Entwickler
Recommender-Module entwickeln	X		Building physical artifacts	Building physical artifacts	Software development	content provider	Entwicklung von Software-Komponenten in Form von Open-Source-Projekten	Recommender-Module sind in der Lage, aus Input-Daten eine Liste von (relevanten) Artefakten zu ermitteln	Kontrolle durch andere Entwickler
Recommender-Konfigurationen definieren	X	X	Building physical artifacts	Building physical artifacts	Creation of configurations	all roles possible	Erstellung von spezifischen Konfigurationen	Recommender-Konfigurationen kombinieren verschiedene Empfehlungsmodulare miteinander	Kontrolle durch andere Benutzer
Informationsvorschläge bewerten	X	X	Evaluating	Evaluating	Rating of information artifacts	perspective provider (explicit)	Bewertung von erhaltenen Informationsartefakten (geeignete/nicht geeignete)	Bewertungsinformationen werden innerhalb der Informationssuche berücksichtigt	Kontrolle durch die Anwendung und das Benutzerprofil
Recommender-Konfigurationen bewerten		X	Evaluating	Evaluating	Rating of recommender configurations	perspective provider (implicit)	Bewertung von Konfigurationen erfolgt implizit über die Bewertung von Informationen	Bewertungsinformationen werden innerhalb der Informationssuche berücksichtigt	Kontrolle durch die Anwendung
Allgemeine Beteiligung	X	X	Kontextabhängige Informationsversorgung in Echtzeit	Kontextabhängige Informationsversorgung in Echtzeit	see previous contributions	all roles possible	Unterschiedliche Beteiligungsformen	Siehe PERCOP-Übersicht	Unterschiedliche Kontrollmechanismen

## A.2: Crowdsourcing-Systeme im WWW

Nature of Collaboration	Architecture	Must recruit users?	What users do?	Examples	Target Problems	Comments
Explicit	Standalone	Yes	Evaluating	<ul style="list-style-type: none"> <li>▶ reviewing and voting at Amazon, tagging Web pages at del.icio.us.com and Google Co-op</li> </ul>	Evaluating a collection of items (e.g., products, users)	Humans as perspective providers. No or loose combination of inputs.
			Sharing	<ul style="list-style-type: none"> <li>▶ Napster, YouTube, Flickr, CPA N, programmableweb.com</li> </ul>	Building a (distributed or central) collection of items that can be shared among users.	Humans as content providers. No or loose combination of inputs.
			<ul style="list-style-type: none"> <li>▶ items</li> <li>▶ textual knowledge</li> <li>▶ structured knowledge</li> </ul>	<ul style="list-style-type: none"> <li>▶ Mailing lists, Yahoo! Answers, Q UIQ, ehow.com, Quora</li> <li>▶ Swivel, Many Eyes, Google Fusion Tables, Google Base, bmr.b.wisc.edu, galaxyzoo, Piazza, Orchestra</li> </ul>		
Implicit	Standalone	Yes	Networking	<ul style="list-style-type: none"> <li>▶ LinkedIn, MySpace, Facebook</li> </ul>	Building social networks	Humans as component providers. Loose combination of inputs.
			Building artifacts	<ul style="list-style-type: none"> <li>▶ Linux, Apache, Hadoop</li> <li>▶ Wikipedia, openmind, Intellipedia, ecolcommunity</li> <li>▶ Wikpedia infoboxes/DBpedia, IWP, Google Fusion Tables, YAGO- NAGA, Cimple/DBLife</li> <li>▶ Wikia Search, mahalo, Freebase, Eurekster</li> <li>▶ newspaper at Digg.com, Second Life</li> </ul>	Building physical artifacts	Humans can play all roles. Typically tight combination of inputs. Some systems ask both humans and machines to contribute.
			Task execution	<ul style="list-style-type: none"> <li>▶ Finding extraterrestrials, elections, finding people, content creation (e.g., Demand Media, Associated Content)</li> </ul>	Possibly any problem	
Implicit	Standalone	Yes	<ul style="list-style-type: none"> <li>▶ play games with a purpose</li> <li>▶ bet on prediction markets</li> <li>▶ use private accounts</li> <li>▶ solve captchas</li> <li>▶ buy/sell/auction, play massive multiplayer games</li> </ul>	<ul style="list-style-type: none"> <li>▶ ESP</li> <li>▶ intrade.com, Iowa Electronic Markets</li> <li>▶ IM DB private accounts</li> <li>▶ recaptchanet</li> <li>▶ eBay, World of Warcraft</li> </ul>	<ul style="list-style-type: none"> <li>▶ labeling images</li> <li>▶ predicting events</li> <li>▶ rating movies</li> <li>▶ digitizing written text</li> <li>▶ building a user community (for purposes such as charging fees, advertising)</li> </ul>	Humans can play all roles. Input combination can be loose or tight.
	Piggyback on another system	No	<ul style="list-style-type: none"> <li>▶ keyword search</li> <li>▶ buy products</li> <li>▶ browse Web sites</li> </ul>	<ul style="list-style-type: none"> <li>▶ Google, Microsoft, Yahoo</li> <li>▶ recommendation feature of Amazon</li> <li>▶ adaptive Web sites (e.g., Yahoo! front page)</li> </ul>	<ul style="list-style-type: none"> <li>▶ spelling correction, epidemic prediction</li> <li>▶ recommending products</li> <li>▶ reorganizing a Web site for better access</li> </ul>	Humans can play all roles. Input combination can be loose or tight.

## A.3: Verbindung von Ontologien innerhalb von PERCOP

